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

KAIST

HyperSearch: Prediction of New Hyperedges through Unconstrained yet Efficient Search





Hyunjin Choo | Fanchen Bu | Hyunjin Hwang | Young-Gyu Yoon | Kijung Shin

Group Interactions are Everywhere!

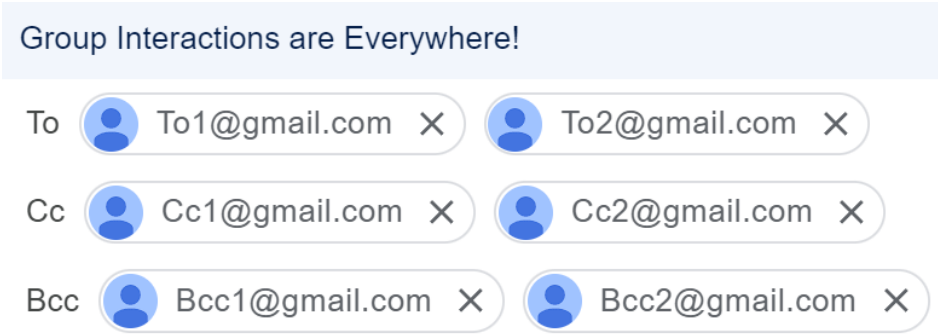
- A group Interaction (GI) is an interaction involving two or more entities

RASP: Robust Mining of Frequent Temporal Sequential Patterns under Temporal Variations

Hyunjin Choo[1], Minho Eom[1], Gyuri Kim[1], Young-Gyu Yoon[1] and Kijung Shin[2*]

[1]School of Electrical Engineering, KAIST, Daejeon, South Korea.
[2]Kim Jaechul Graduate School of AI, KAIST, Seoul, South Korea.

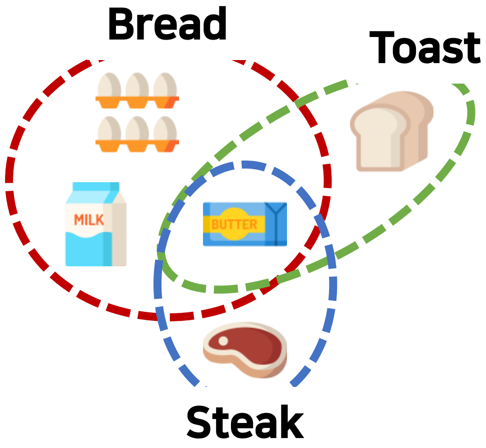
Co-authorship



Email addresses in an email

#boot
#networking
#drivers
#server
#wireless

Tags added to a question



Ingredients in recipes

Hypergraphs Model Group Interactions

- Hypergraphs offer a natural framework for modeling group interactions

RASP: Robust Mining of Frequent Temporal Sequential Patterns under Temporal Variations

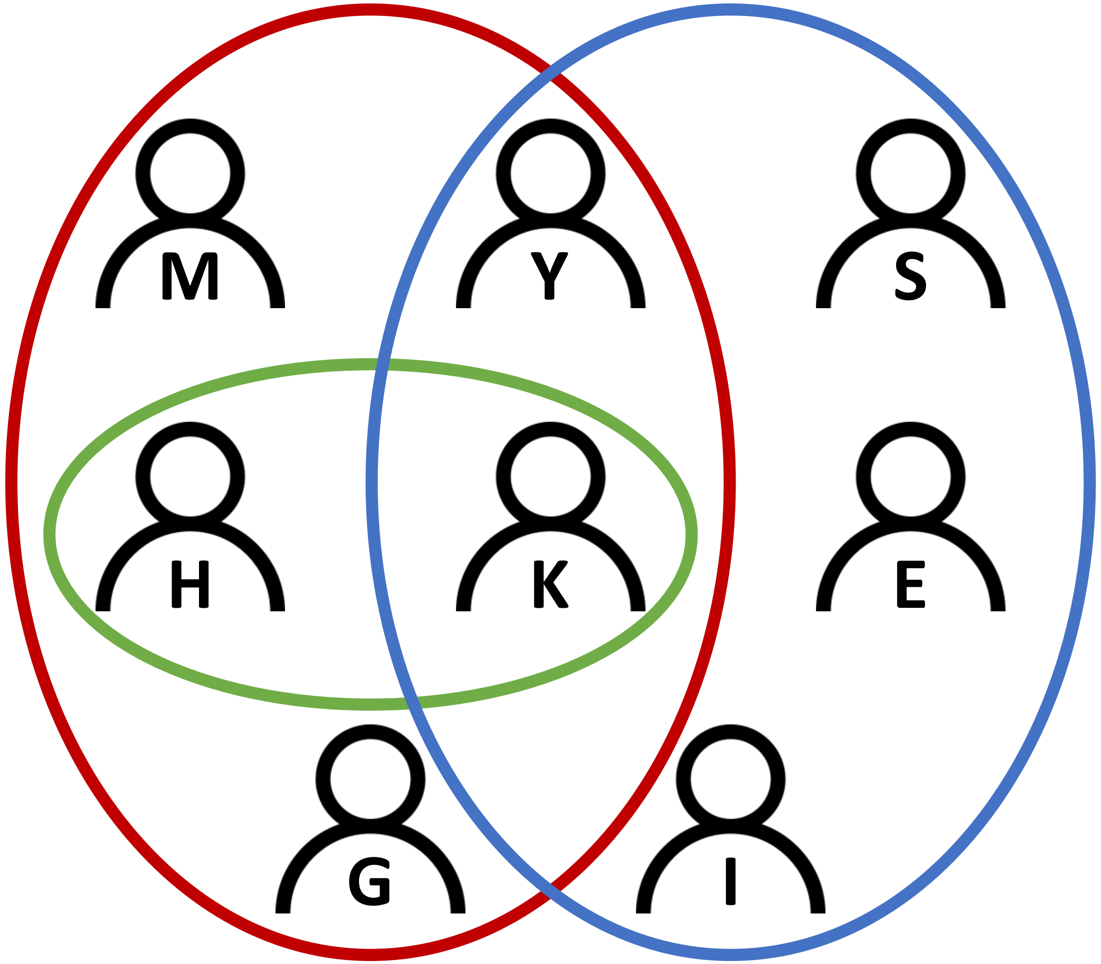
Hyunjin Choo[1], Minho Eom[1], Gyuri Kim[1], Young-Gyu Yoon[1] and Kijung Shin[2*]

On the Persistence of Higher-Order Interactions in Real-World Hypergraphs

Hyunjin Choo* Kijung Shin†

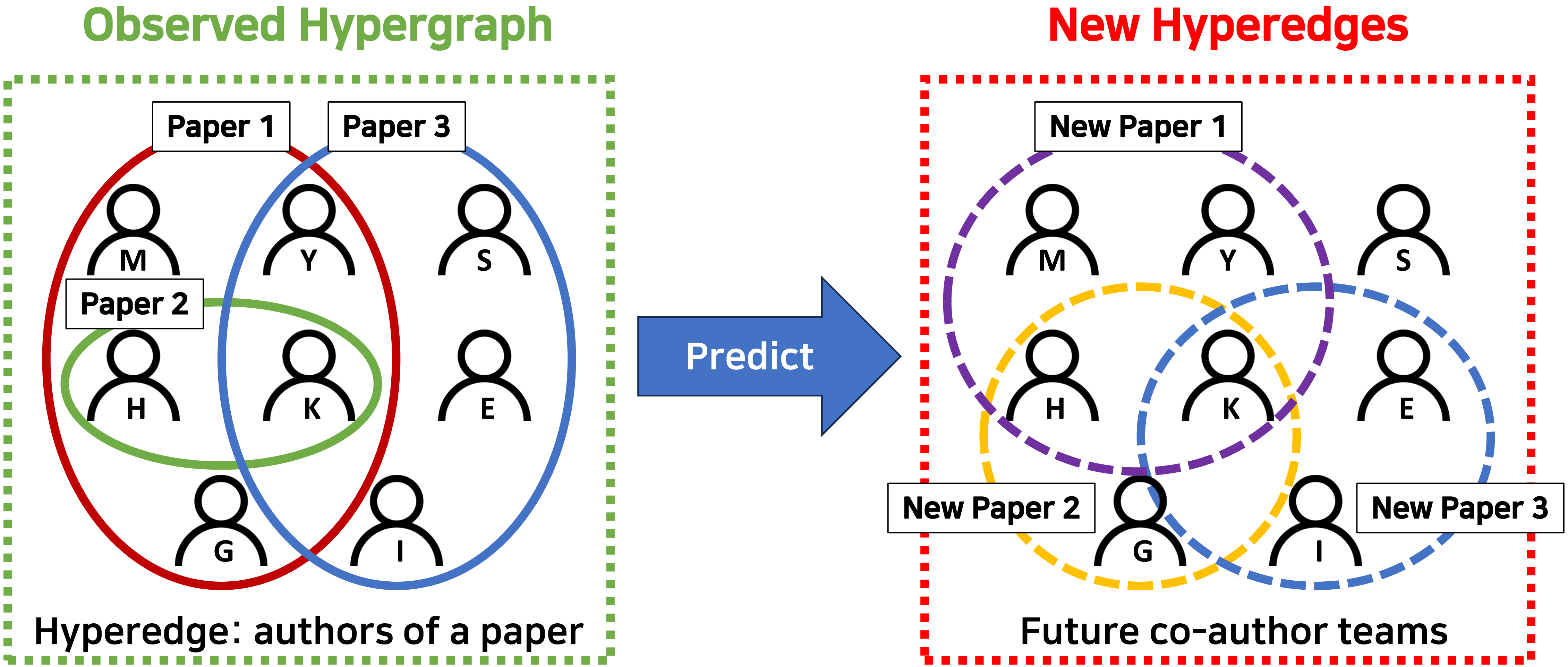
Efficient Neural Network Approximation of Robust PCA for Automated Analysis of Calcium Imaging Data

Seungjae Han[1], Eun-Seo Cho[1], Inkyu Park[2], Kijung Shin[1,2], and Young-Gyu Yoon[1]



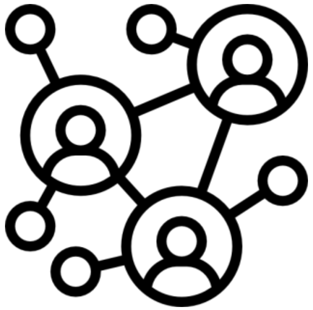
Problem: Hyperedge Prediction

- Given an observed hypergraph, predict new (future) hyperedges



Real-World Applications

- Group recommendation [Liben-Nowell et al., 2003, Wang, Peng, et al, 2015]
  - Recommending relevant groups within social networks enhances user experiences
- Collaboration prediction [Wang, Xi et al., 2014, Lande, et al., 2020]
  - Predicting collaborations with shared interests or expertise optimizes team formation
- Drug discovery [Jin, Shuting, et al., 2023, Saifuddin, K. M., et al., 2023]
  - Forecasting functional groups of protein complexes or genes facilitates drug discovery



Group Recommendation

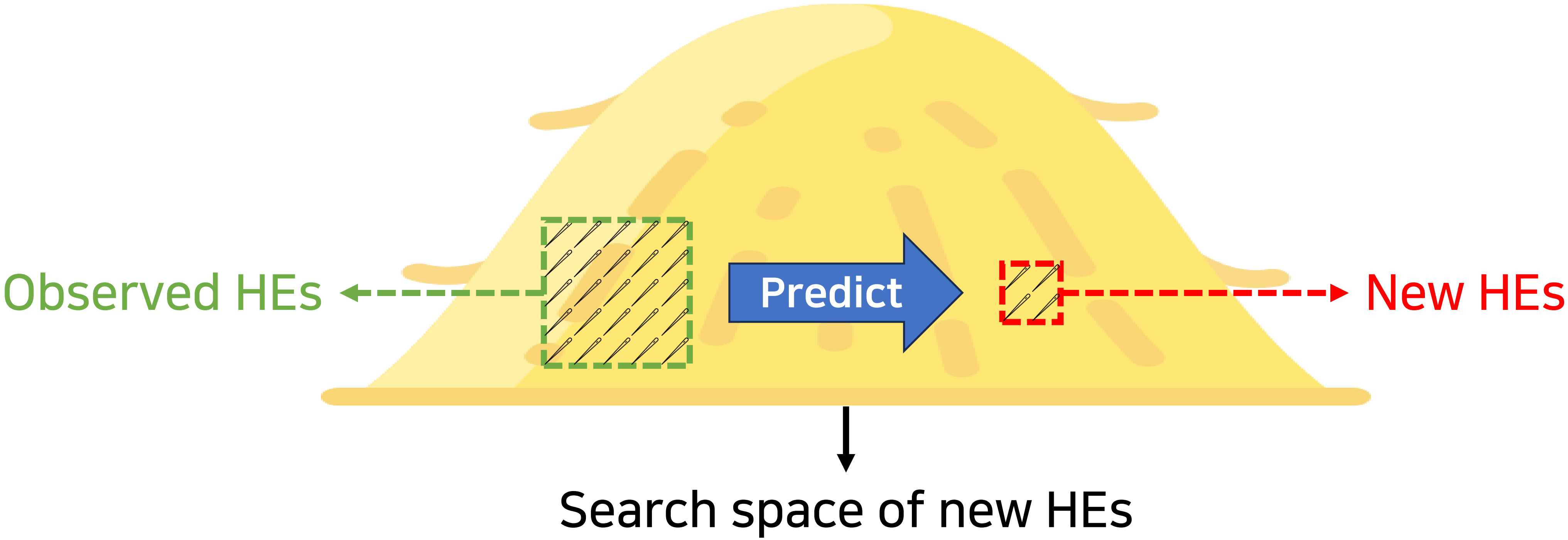
Collaboration prediction

Drug Discovery

Challenge: Vast Search Space of New Hyperedges

- Search space of new hyperedges: $\mathcal{O}(2^n)$ for n nodes
  - E.g., In DBLP, $n = 15{,}639$ → Search Space of new HEs: $2^{15{,}639} = 6.4 \times 10^{4{,}707}$

"Finding a needle in a haystack"



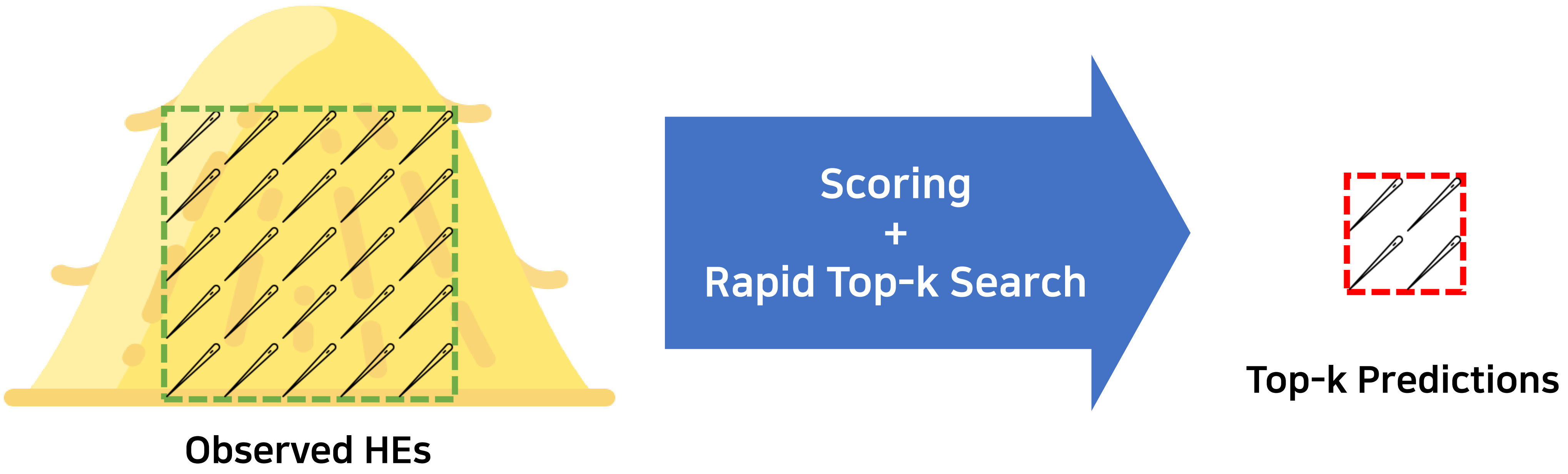
Related Works and Contributions

- **Limitations of prior works**
  - Constrained candidate sets
    - Limited to binary classification, distinguishing between positive and negative within candidate set
    - How to obtain a promising candidate set with the ground-truth is not addressed
  - Unjustified structural assumptions
    - Methods that avoid candidate sets often depend on structural assumptions

- **Our contributions**
  - We proposed a principled and learnable algorithm without requiring a candidate set
  - HyperSearch directly generates a candidate set, which is much smaller than search space

HyperSearch: Overview

- **Goal**: Predict new hyperedges from a vast search space
  - **Component 1**: Scoring based on empirical observations
  - **Component 2**: Rapid top-k search with pruning



Hyperedge Scoring Based on Observations

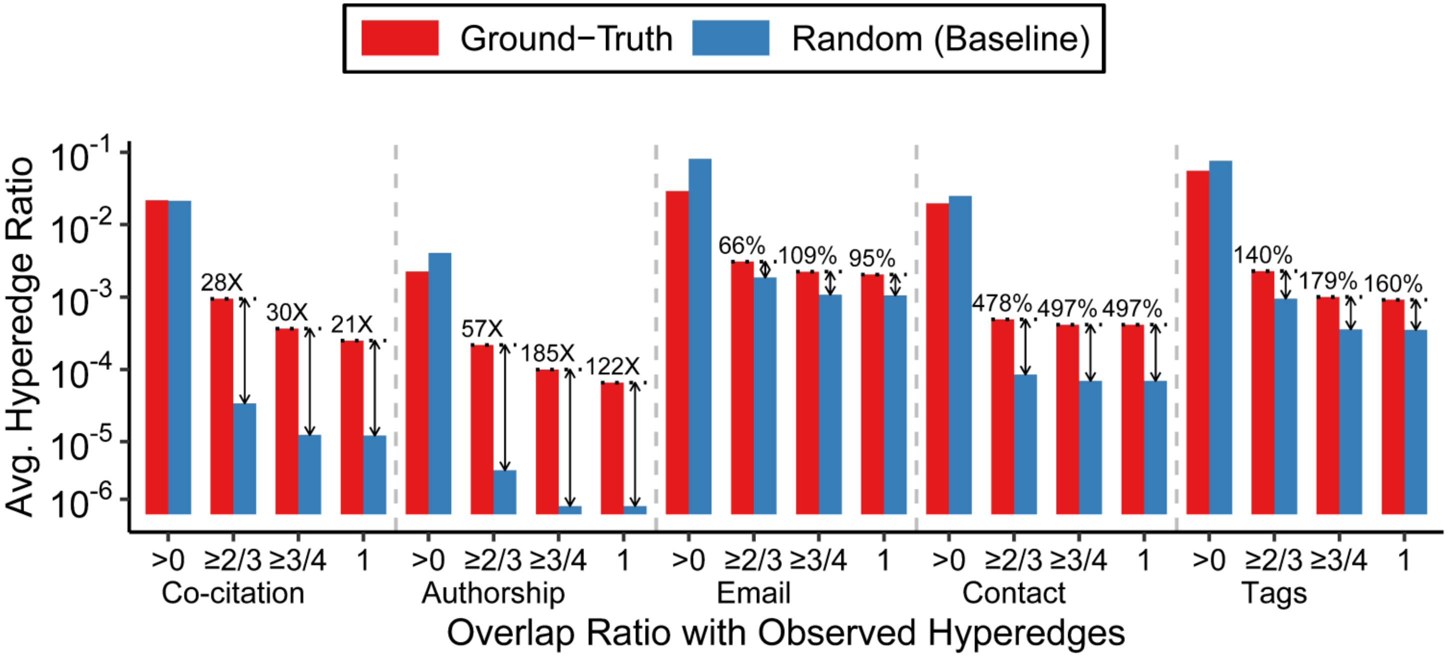
- **Goal**: Predict new hyperedges from a vast search space
  - **Component 1**: Scoring based on empirical observations

1. Significant Overlap between hyperedges
2. Temporal Bias in Structural Overlap

  - **Component 2**: Rapid top-k search with pruning

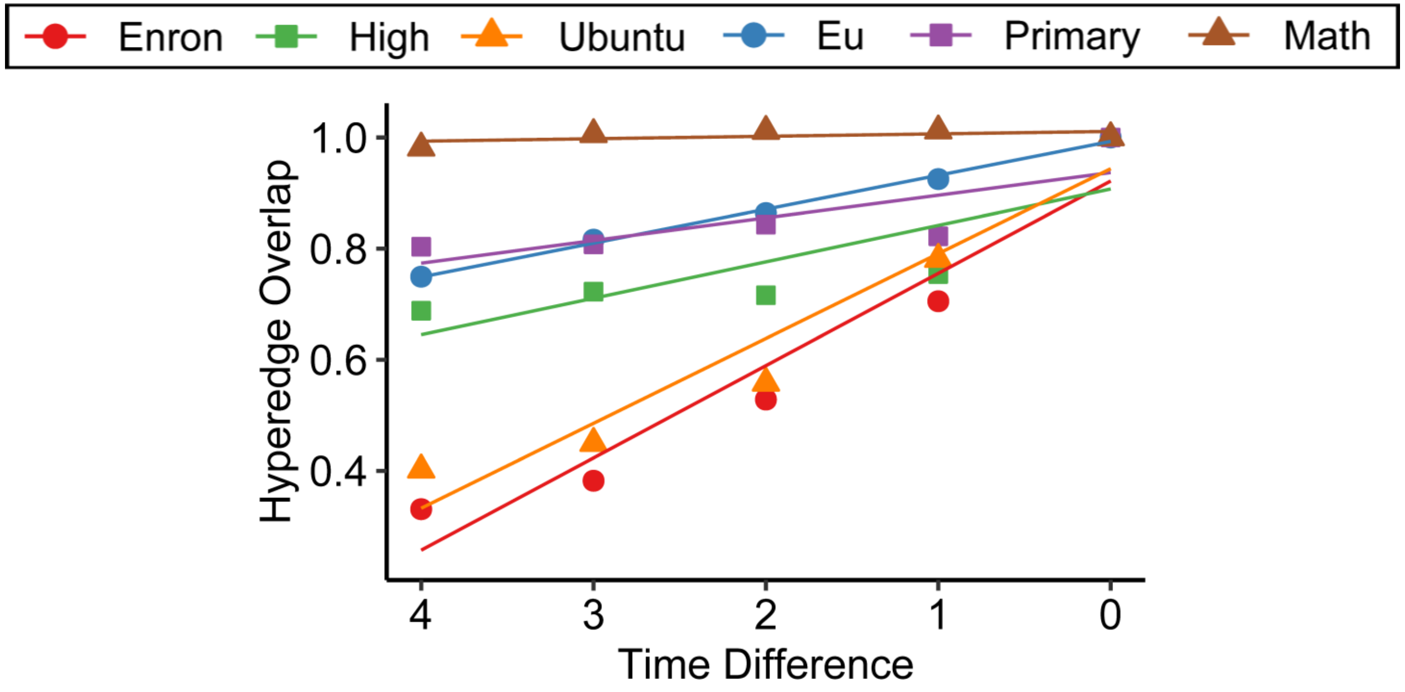
Obs. 1: Significant Overlap between Hyperedges

- We measure the structural overlap between new (20%) and observed hyperedges (80%), and compare it with random hyperedges as a baseline
  - High-overlap hyperedges are more frequent in ground truth than random
  - ➢ New hyperedges are more likely to substantially overlap with existing ones.
  - ➢ Scoring function prioritizes candidates with high overlap



Obs. 2: Temporal Bias in Structural Overlap

- We measure the structural overlap between new (20%) and observed hyperedges (80%) across different timestamp groups, using five equal-sized partitions
  - Overlap increases as the time gap between new and observed groups decreases
  - ➢ New hyperedges tend to reuse more recent existing interactions than earlier ones
  - ➢ Scoring function adds higher weights to more recent observed hyperedges



Component 1: Scoring Based on Observations

- **S1**. Prioritizing candidates with high overlap
  - Obs. 1: Significant overlap between hyperedges

- **S2**. Weighting more recent observed hyperedges
  - Obs. 2: Temporal bias in structural overlap

S1. Concepts: Relaxed Overlap Count

- Inspired by *support* in frequent itemset mining:
  - Support counts co-appearance of nodes across observed hyperedges
  - Exact co-appearance is too strict → allow partial co-appearance
- Keep only eligible supporters that satisfy three relaxation criteria → subset $\tilde{E} \subset E$
  - Node (ϵ_v): each candidate node is present in most supporters
  - Hyperedge (ϵ_e): any single supporter may miss only a small fraction of hyperedge candidate e'
  - Total (ϵ_t): overall missing occurrences remain limited
- **Relaxed overlap count**: $ovr(e', \epsilon_v, \epsilon_e, \epsilon_t) = |\tilde{E}(e', \epsilon_v, \epsilon_e, \epsilon_t)|$

Node (columns) / **Hyperedge** (rows)

	v_1	v_2	v_3
e_1	1	1	0
e_2	1	0	1
e_3	0	1	1

→

$e' = \{v_1, v_2, v_3\}$

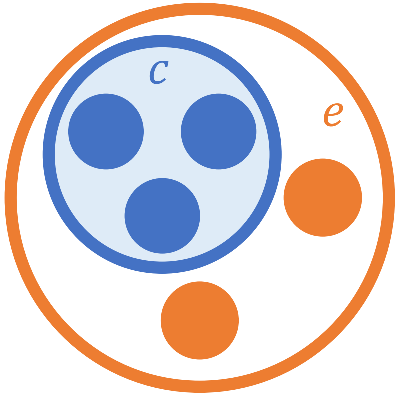
Support of e': 0

$ovr(e', 1/3, 1/3, 1/3) = 3$

S1. Incorporating Overlap Ratio

- Relaxed overlap count only accounts for the number of observed hyperedges that satisfy the relaxation criteria
  - We further incorporate overlap ratio to capture the degree of overlap

$$OverlapRatio(c, e) \coloneqq \frac{|c \cap e|}{|e|}$$



$$OverlapRatio(c, e) = 3/5$$

S2. Time Weight

- **Time weight**: assigns greater significance to more recent hyperedges

$$\exp(\tau t_e)$$

  - τ: adjustable parameter that determines the emphasis on recent hyperedges
  - $t_e \in [0,1]$: normalized timestamp of e

Final Scoring Function of HyperSearch

- Final score for a hyperedge candidate c :

$$f_s(c) = \sum_{\underbrace{e\in\tilde{E}(c,\epsilon_v,\epsilon_e,\epsilon_t)}_{\text{Relaxation Criteria}}} \underbrace{\frac{|c\cap e|}{|e|}}_{\text{Overlap Ratio}} \underbrace{\exp(\tau t_e)}_{\text{Time Weight}}$$

Component 2: Rapid Top-k Search with Pruning

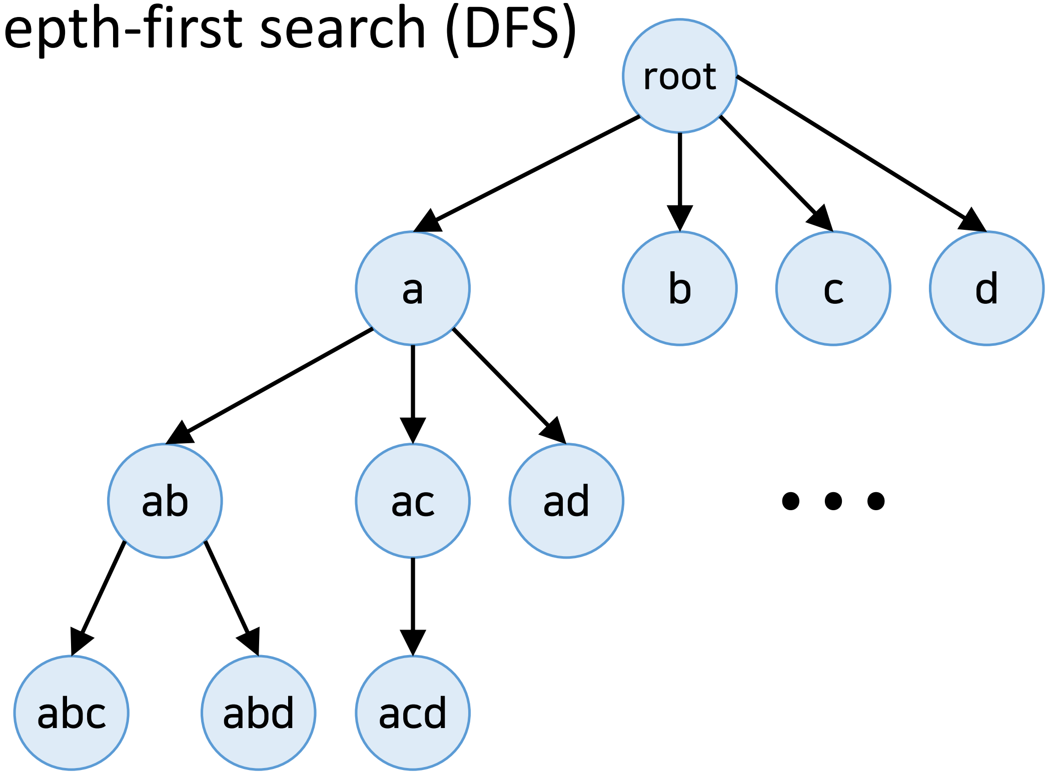
- **Goal**: Predict new hyperedges from a vast search space
  - **Component 1**: Scoring based on empirical observations

  - **Component 2**: Rapid top-k search with pruning

1. Search strategy
2. Pruning scheme
3. Top-k selection

Overview of Top-k Search (1)

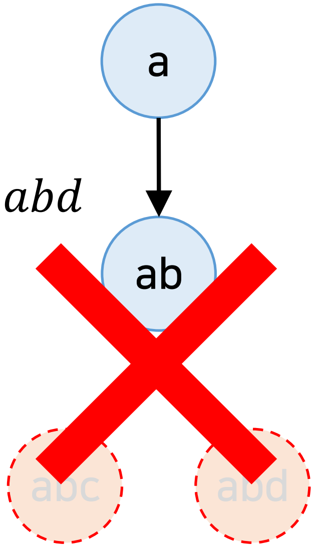
- **Given** observed hyperedges,
- Top-k search aims to identify the top-k highest-scoring new hyperedges
    - **1. Search strategy**
        - Employ a depth-first search (DFS)



Overview of Top-k Search (2)

- **Given** observed hyperedges,
- Top-k search aims to identify the top-k highest-scoring new hyperedges
  - 1. Search strategy
  - **2. Pruning scheme**
    - If a scoring function is anti-monotonic, search space can be pruned by a threshold θ
    - If $f_s(S) < \theta$, we can prune all its supersets together

$f_s(ab) < \theta$
$\Rightarrow$ Prune ab, abc, abd

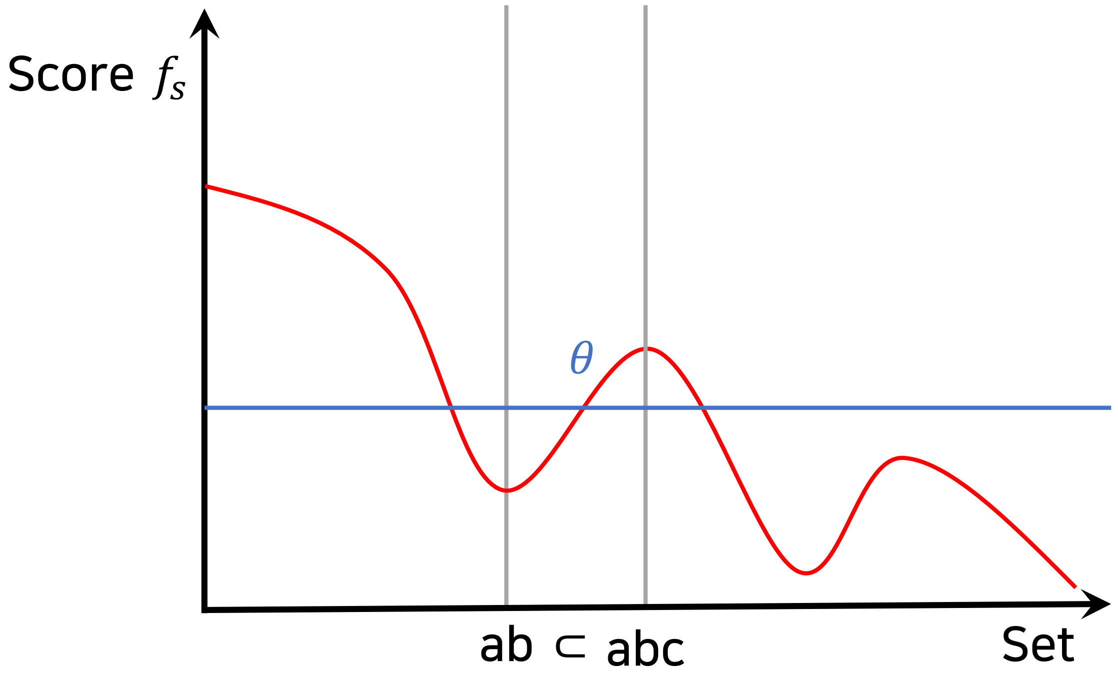


Overview of Top-k Search (2)

- **Given** observed hyperedges,
- Top-k search aims to identify the top-k highest-scoring new hyperedges
  - 1. Search strategy
  - **2. Pruning scheme**
    - If a scoring function is anti-monotonic, search space can be pruned by a threshold θ
    - However, our scoring function f_s is not anti-monotonic for arbitrary relaxation ratios
    - We use an anti-monotonic upper bound function f_n instead

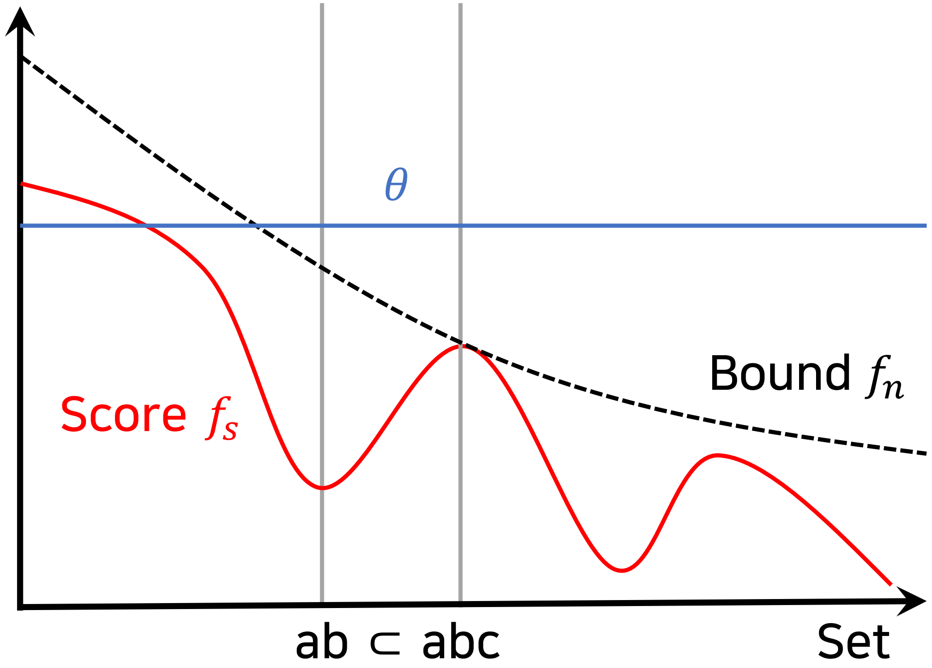
Details Scoring Function is Not Anti-Monotonic

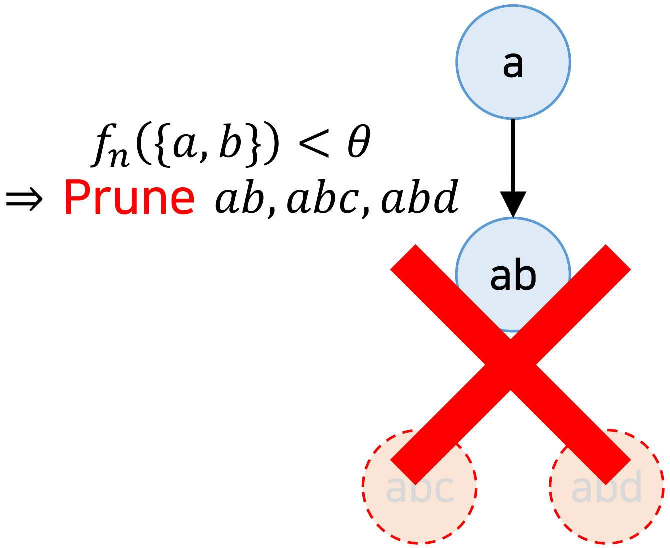
- Our scoring function f_s is not anti-monotonic
  - $\{a, b\} \subset \{a, b, c\} \nRightarrow f_s(\{a, b\}) \geq f_s(\{a, b, c\})$
  - ➢ $f_s(\{a, b\}) < \theta \nRightarrow f_s(\{a, b, c\}) < \theta \Rightarrow$ all supersets of $\{a, b\}$ cannot be pruned



Details Anti-Monotonic Upper-Bound Function

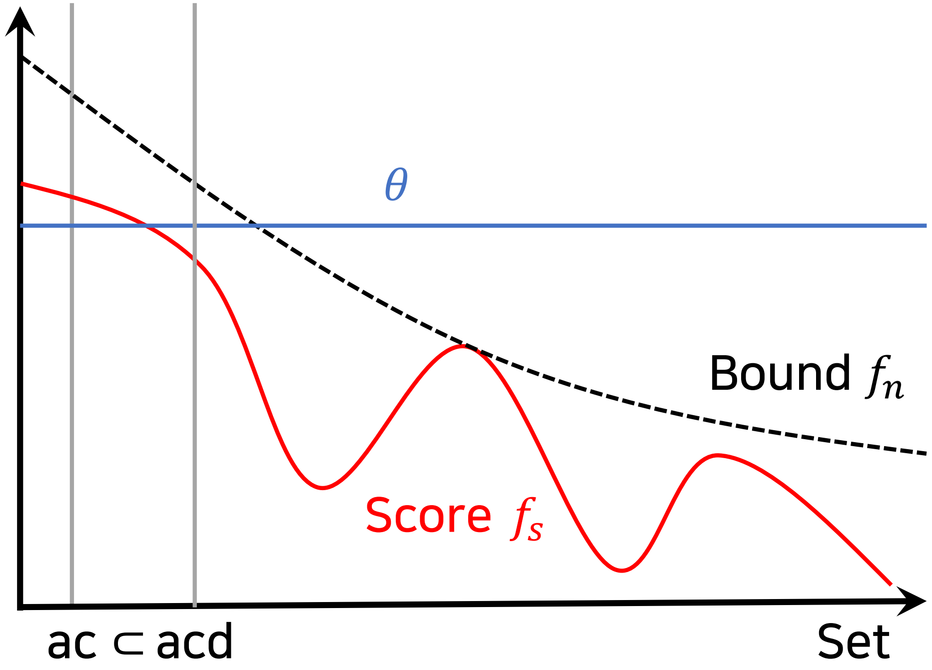
- Search space can be bounded on an anti-monotonic upper-bound function f_n
  - $f_n(S) < \theta \Rightarrow$ all supersets of S can be pruned from the search space





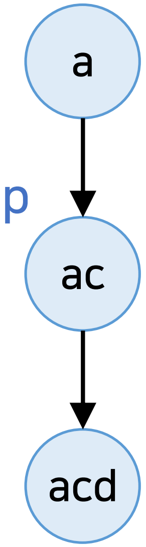
Details Anti-Monotonic Upper-Bound Function

- Search space can be bounded on an anti-monotonic upper-bound function f_n
  - $f_n(S) \geq \theta \Rightarrow$ proceed to next step



$f_n(\{a, c\}) \geq \theta$

$\Rightarrow$ Proceed to next step



Overview of Top-k Search (3)

- **Given** observed hyperedges,
- Top-k search aims to identify the top-k highest-scoring new hyperedges
  - 1. Search strategy
  - 2. Pruning scheme
  - **3. Top-k selection**
    - Select the top-k new hyperedge candidates based on the scoring function

Datasets

- 10 Real-world datasets from 5 domains:
  - Non-temporal hypergraphs
    - **Co-citation**: Groups of cited papers in papers (Citeseer, Cora)
    - **Authorship**: Groups of papers by authors (Cora-A, DBLP-A)
  - Temporal hypergraphs
    - **Contact**: Groups of people in contact (High, Primary)
    - **Email**: Groups of email addresses on emails (Enron, Eu)
    - **Tags**: Groups of tags attached to questions (Math.sx, Ubuntu)

Contact

Email

Tags

Co-citation

Authorship

Experimental Settings

- Hyperedge splits in datasets
  - Observed 80% : New 20%
  - Temporal hypergraphs: Old vs. recent
  - Non-temporal hypergraphs: 5 random splits (results are averaged)

- Evaluation measure for accuracy
  - **Recall@$\boldsymbol{k}$**: How many true HEs were correctly predicted?

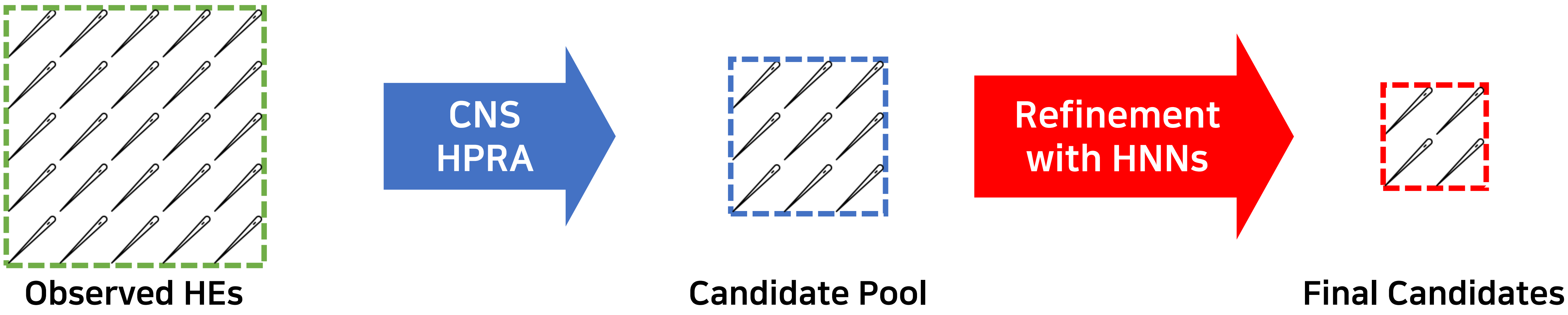
$$\frac{|Predicted\ HEs\ \cap True\ HEs|}{|True\ HEs|}$$

- $\boldsymbol{k}$: Target number of outcomes (i.e., candidates): $\{1, 2, 5\} \times |True\ HEs|$

Baseline Methods

- **1 Stage only**
  - **Clique negative sampling (CNS)** [Patil et al., 2020]: Pick a random hyperedge and replace a random node with an adjacent one
  - **HPRA** [Kumar et al., 2020]: Hyperedge prediction using resource allocation

- **2 Stages**
  - CNS or HPRA → Refinement with hypergraph neural networks (HNNs)



Q1. HyperSearch is Accurate

- Across all non-temporal hypergraph settings, HyperSearch performs best

: The best methods : The second-best methods

Dataset	Citeseer			Cora			Cora-A			DBLP-A		
Method (↓) / $\mathcal{K}$ (→)	1×	2×	5×	1×	2×	5×	1×	2×	5×	1×	2×	5×
HyperSearch (Proposed)	8.2 (1.6)	10.9 (1.5	17.9 (1.8)	7.5 (1.8)	10.0 (2.0)	14.6 (1.5)	7.3 (3.6)	10.9 (2.5)	16.4 (2.9)	5.4 (0.1)	8.4 (0.2)	14.3 (0.4)
CNS	1.5 (0.2)	3.3 (0.8)	8.8 (1.4)	2.9 (2.1)	5.9 (1.5)	12.5 (2.1)	0.3 (0.2)	0.6 (0.6)	2.1 (0.8)	0.7 (0.2)	1.2 (0.1)	2.7 (0.2)
HPRA	0.2 (0.4)	0.3 (0.4)	0.8 (0.6)	0.2 (0.2)	0.6 (0.5)	2.3 (1.5)	0.0 (0.0)	0.1 (0.2)	0.1 (0.2)	0.0 (0.0)	0.0 (0.0)	0.1 (0.0)
MHP	2.8 (1.1)	4.4 (1.3)	8.9 (1.4)	1.2 (0.9)	2.4 (1.1)	6.0 (1.6)	0.8 (0.2)	1.6 (0.2)	6.1 (2.8)	-	-	-
MHP-C	2.3 (1.0)	5.7 (1.7)	-	4.2 (1.3)	8.0 (1.5)	-	0.4 (0.4)	1.4 (0.7)	2.6 (0.5)	-	-	-
AHP-C	2.4 (0.9)	5.2 (1.2)	-	4.0 (1.0)	8.5 (1.8)	-	0.4 (0.4)	0.9 (0.6)	1.7 (0.7)	-	-	-
SAGNN-C	1.8 (0.6)	4.3 (1.4)	-	3.8 (1.7)	7.5 (2.2)	-	0.3 (0.3)	0.7 (0.5)	1.5 (0.6)	0.7 (0.1)	1.2 (0.2)	2.3 (0.4)
NHP-C	2.3 (0.9)	5.5 (1.2)	-	4.2 (1.3)	7.4 (1.2)	-	0.4 (0.3)	0.9 (0.3)	2.2 (0.6)	0.9 (0.2)	1.6 (0.2)	3.4 (0.2)
MHP-H	0.3 (0.4)	0.7 (0.6)	-	0.6 (0.5)	1.9 (1.2)	3.4 (1.4)	0.1 (0.1)	0.1 (0.1)	0.1 (0.1)	-	-	-
AHP-H	0.0 (0.0)	0.1 (0.1)	-	0.5 (0.0)	1.4 (0.0)	1.8 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	-	-	-
SAGNN-H	0.2 (0.2)	0.4 (0.3)	-	0.4 (0.4)	1.2 (0.8)	2.1 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	-
NHP-H	0.1 (0.2)	0.3 (0.3)	-	0.6 (0.5)	1.9 (1.2)	3.4 (1.5)	0.1 (0.2)	0.1 (0.2)	0.1 (0.2)	0.0 (0.0)	0.0 (0.0)	-

-: out-of-time (> 2 days).

Q1. HyperSearch is Accurate

- Across most temporal hypergraph settings, HyperSearch performs best

: The best methods : The second-best methods

Dataset	Enron			Eu			High			Primary			Ubuntu			Math-sx		
Method (↓) / $\mathcal{K}$ (→)	1×	2×	5×	1×	2×	5×	1×	2×	5×	1×	2×	5×	1×	2×	5×	1×	2×	5×
HyperSearch (Proposed)	16.1	25.6	33.1	12.4	17.3	26.8	14.8	18.3	27.3	7.3	11.8	20.8	12.0	15.4	20.6	12.1	17.3	24.5
CNS	10.3	16.4	29.7	5.1	10.9	22.0	12.6	13.8	18.1	4.5	7.3	11.9	1.6	2.9	6.7	3.4	6.0	11.7
HPRA	1.7	5.8	9.2	3.5	5.8	10.2	9.0	14.8	28.1	4.7	8.1	20.4	1.1	2.0	4.4	2.2	3.9	8.1
MHP	0.3	0.6	3.6	0.3	1.0	3.4	0.9	2.9	7.4	4.3	7.7	21.1	-	-	-	-	-	-
MHP-C	7.6	14.9	22.0	7.4	14.1	22.7	4.3	5.7	8.3	4.1	5.7	9.6	-	-	-	-	-	-
SAGNN-C	6.0	8.2	14.6	8.6	16.0	24.5	7.2	8.5	9.9	5.1	7.9	11.9	2.3	4.3	8.7	4.7	7.9	14.5
NHP-C	13.3	20.3	29.8	8.1	14.9	23.3	9.7	11.6	13.7	5.3	8.5	13.1	1.8	3.5	7.1	3.6	6.1	11.2
MHP-H	4.6	5.4	9.5	4.4	6.5	14.2	9.1	16.3	31.8	5.4	11.4	22.6	-	-	-	-	-	-
SAGNN-H	2.8	3.4	7.1	5.4	8.1	17.8	10.2	18.1	34.1	4.5	9.4	19.5	2.0	3.5	-	3.7	6.5	-
NHP-H	4.5	5.2	9.5	4.6	6.7	14.2	12.4	16.6	32.7	6.0	11.7	22.5	1.6	2.8	-	2.6	4.6	-

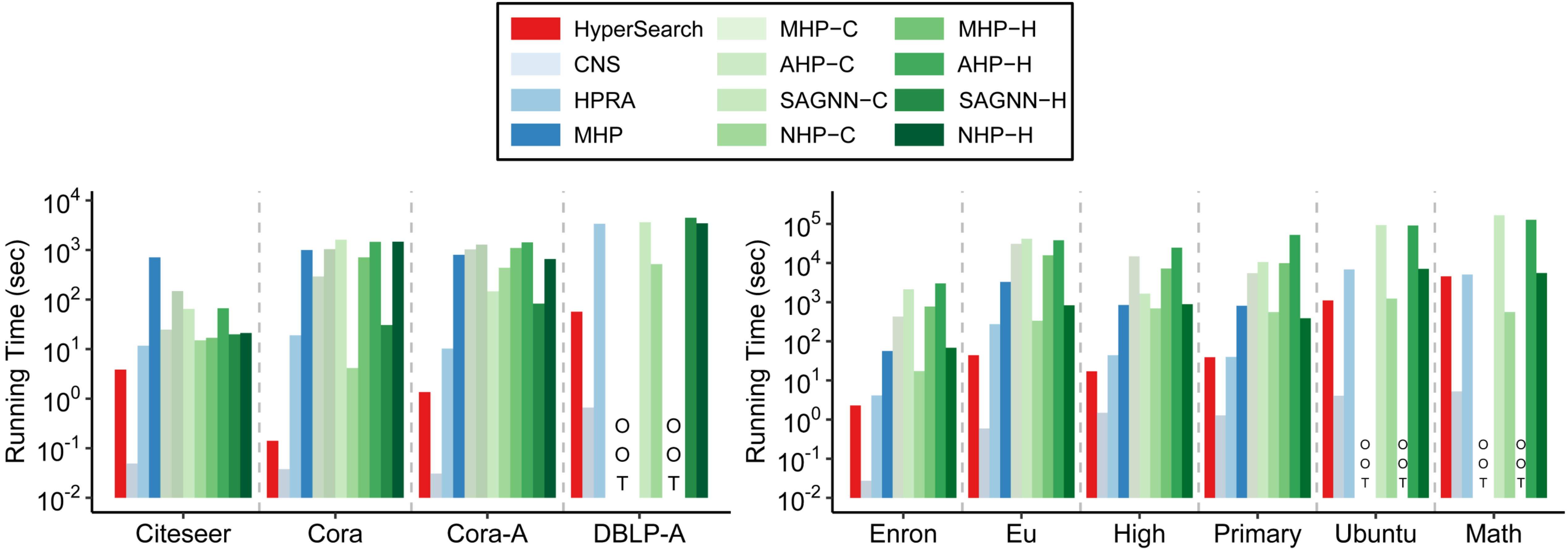
-: out-of-time (> 2 days).

Q1. Case Studies: Semantically Coherent Predictions

- Top-scoring predicted hyperedges (size 2–5) on Tags datasets
  - **Math examples**:
    - [ring-theory, noetherian]
    - [matrices, vectors, vector-spaces]
    - [group-theory, finite-groups, fieldtheory, abstract-algebra]
    - [calculus, sequences-and-series, real-analysis, integration, convergence]
  - **Ubuntu examples**:
    - [drivers, xorg]
    - [boot, grub2, btrfs]
    - [dual-boot, boot, live-usb, grub2]
    - [partitioning, grub2, 16.04, dual-boot, boot]

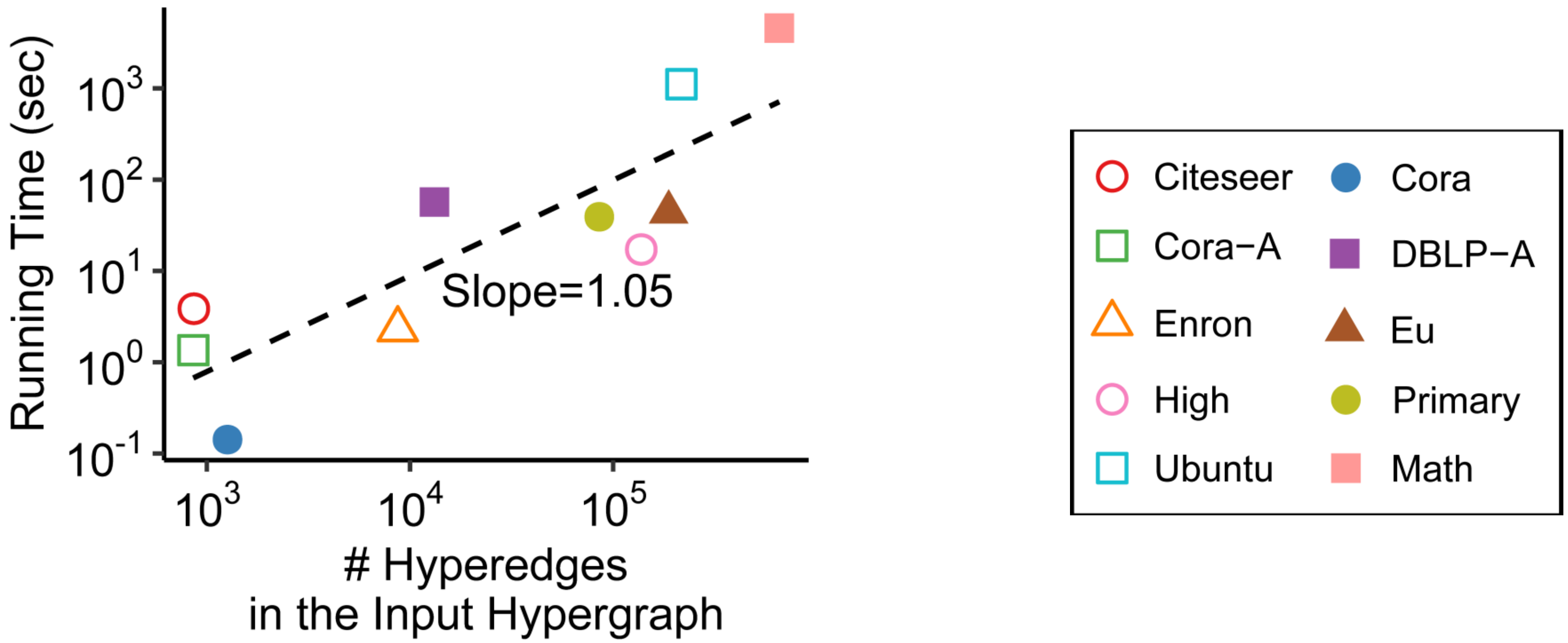
Q2. HyperSearch is Fast

- HyperSearch runs faster than deep learning-based methods in most cases



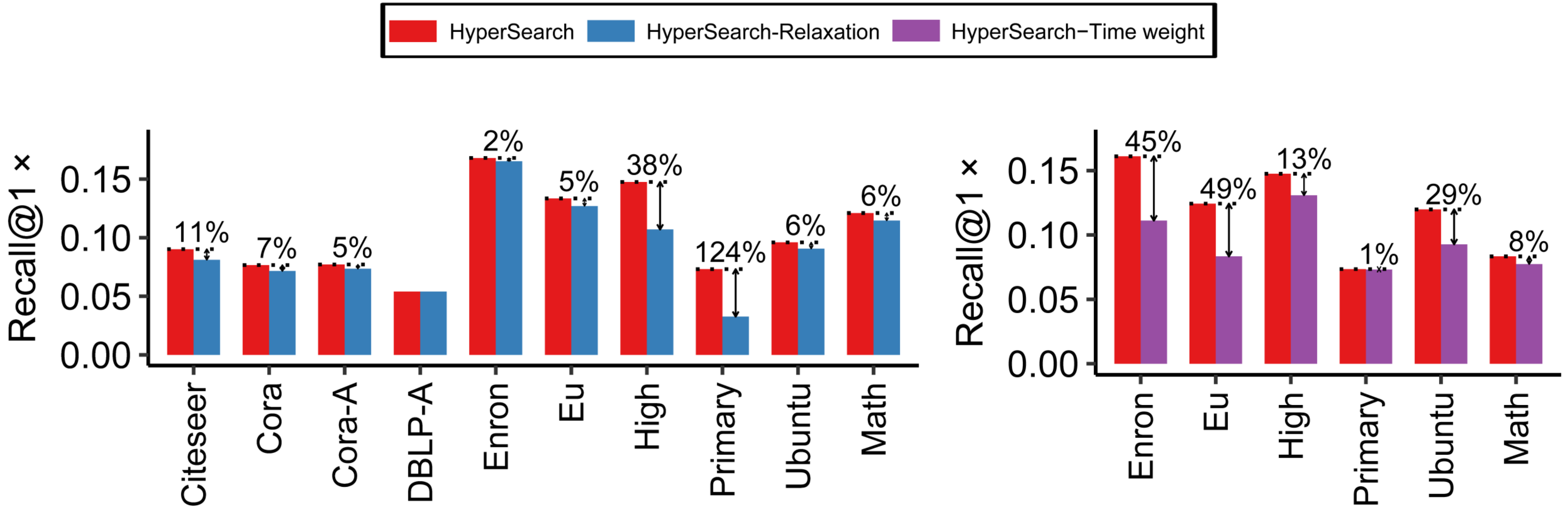
Q2. HyperSearch is Scalable

- Runtime of HyperSearch scales almost linearly with the number of hyperedges



Q3. Each Component Contributes to its Performance

- In most cases, HyperSearch outperforms its variants with missing components



Conclusion

- We proposed HyperSearch to predict new hyperedges from a vast search space

Observations

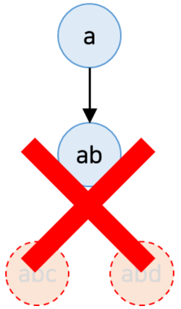
(1) Overlap between HEs
(2) Temporal Bias

Scoring

Accurate and Efficient Search

	v_1	v_2	v_3
e_1	1	1	0
e_2	1	0	1
e_3	0	1	1

Relaxed Concepts

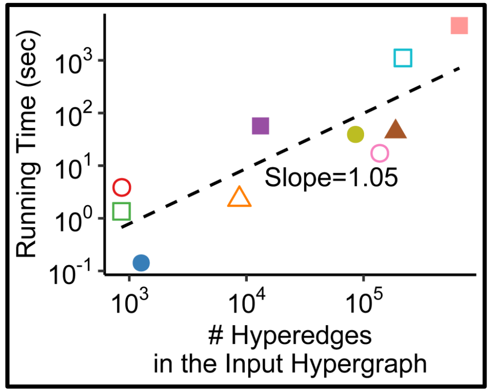


Pruning Scheme

Strong Performance

Dataset	Citeseer			Cora			Cora-A			DBLP-A		
Method (↓) / $\mathcal{K}$ (→)	1×	2×	5×	1×	2×	5×	1×	2×	5×	1×	2×	5×
HyperSearch (Proposed)	8.2 (1.6)	10.9 (1.5	17.9 (1.8)	7.5 (1.8)	10.0 (2.0)	14.6 (1.5)	7.3 (3.6)	10.9 (2.5)	16.4 (2.9)	5.4 (0.1)	8.4 (0.2)	14.3 (0.4)
CNS	1.5 (0.2)	3.3 (0.8)	8.8 (1.4)	2.9 (2.1)	5.9 (1.5)	12.5 (2.1)	0.3 (0.2)	0.6 (0.6)	2.1 (0.8)	0.7 (0.2)	1.2 (0.1)	2.7 (0.2)
HPRA	0.2 (0.4)	0.3 (0.4)	0.8 (0.6)	0.2 (0.2)	0.6 (0.5)	2.3 (1.5)	0.0 (0.0)	0.1 (0.2)	0.1 (0.2)	0.0 (0.0)	0.0 (0.0)	0.1 (0.0)
MHP	2.8 (1.1)	4.4 (1.3)	8.9 (1.4)	1.2 (0.9)	2.4 (1.1)	6.0 (1.6)	0.8 (0.2)	1.6 (0.2)	6.1 (2.8)	-	-	-
MHP-C	2.3 (1.0)	5.7 (1.7)	-	4.2 (1.3)	8.0 (1.5)	-	0.4 (0.4)	1.4 (0.7)	2.6 (0.5)	-	-	-
AHP-C	2.4 (0.9)	5.2 (1.2)	-	4.0 (1.0)	8.5 (1.8)	-	0.4 (0.4)	0.9 (0.6)	1.7 (0.7)	-	-	-
SAGNN-C	1.8 (0.6)	4.3 (1.4)	-	3.8 (1.7)	7.5 (2.2)	-	0.3 (0.3)	0.7 (0.5)	1.5 (0.6)	0.7 (0.1)	1.2 (0.2)	2.3 (0.4)
NHP-C	2.3 (0.9)	5.5 (1.2)	-	4.2 (1.3)	7.4 (1.2)	-	0.4 (0.3)	0.9 (0.3)	2.2 (0.6)	0.9 (0.2)	1.6 (0.2)	3.4 (0.2)
MHP-H	0.3 (0.4)	0.7 (0.6)	-	0.6 (0.5)	1.9 (1.2)	3.4 (1.4)	0.1 (0.1)	0.1 (0.1)	0.1 (0.1)	-	-	-
AHP-H	0.0 (0.0)	0.1 (0.1)	-	0.5 (0.0)	1.4 (0.0)	1.8 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	-	-	-
SAGNN-H	0.2 (0.2)	0.4 (0.3)	-	0.4 (0.4)	1.2 (0.8)	2.1 (1.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	-
NHP-H	0.1 (0.2)	0.3 (0.3)	-	0.6 (0.5)	1.9 (1.2)	3.4 (1.5)	0.1 (0.2)	0.1 (0.2)	0.1 (0.2)	0.0 (0.0)	0.0 (0.0)	-

-: out-of-time (> 2 days).



Source code and datasets are available at https://github.com/jin-choo/HyperSearch/

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

KAIST

HyperSearch: Prediction of New Hyperedges through Unconstrained yet Efficient Search

Hyunjin Choo | Fanchen Bu | Hyunjin Hwang | Young-Gyu Yoon | Kijung Shin

References

[Liben-Nowell et al., 2003] Liben-Nowell, David, and Jon Kleinberg. "The link prediction problem for social networks." CIKM 2003

[Wang, Peng, et al, 2015] Wang, Peng, et al. "Link prediction in social networks: the state-of-the-art." Science China Information Sciences (2015)

[Wang, Xi et al., 2014] Wang, Xi, and Gita Sukthankar. "Link prediction in heterogeneous collaboration networks." Social network analysis-community detection and evolution (2014)

[Lande, et al., 2020] Lande, Dmytro, et al. "Link prediction of scientific collaboration networks based on information retrieval." WWW 2020

[Jin, Shuting, et al., 2023] Jin, Shuting, et al. "A general hypergraph learning algorithm for drug multi-task predictions in micro-to-macro biomedical networks." PLOS Computational Biology (2023)

[Saifuddin, K. M., et al., 2023] Saifuddin, K. M., et al. "Hygnn: Drug-drug interaction prediction via hypergraph neural network." ICDE 2023

[Hwang et al., 2022] H. Hwang et al., “Ahp: Learning to negative sample for hyperedge prediction”, SIGIR 2022

[Yu et al., 2024] H. Yu et al., “Prediction Is NOT Classification: On Formulation and Evaluation of Hyperedge Prediction”, HDM 2024

[Zhang, Ruochi et al., 2020] Zhang, Ruochi et al., “Hyper-SAGNN: a self-attention based graph neural network for hypergraphs.” ICLR 2020

[Yadati et al., 2020] Yadati, Naganand et al., “Nhp: Neural hypergraph link prediction.” CIKM 2020

[Patil et al., 2020] P. Patil et al., “Negative sampling for hyperlink prediction in networks”, PAKDD 2020

[Kumar et al., 2020] T. Kumar et al., “HPRA: Hyperedge prediction using resource allocation”, WebSci 2020