Improving the Core Resilience of Real-world Hypergraphs

Manh Tuan Do, Kijung Shin

{manh.it97, kijungs}@kaist.ac.kr

Code & Data: *https://github.com/manhtuando97/CoReA*

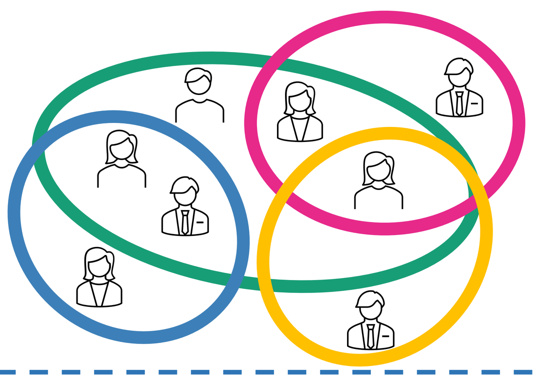
Summary

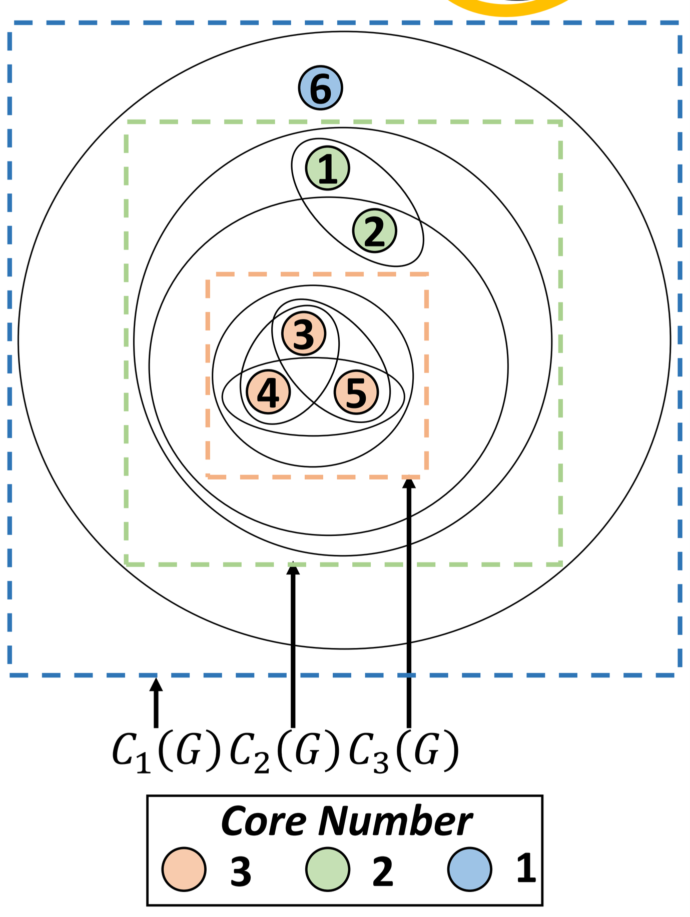
- **Novel Problem:** Improvement of core resilience in hypergraphs.
- **Concepts & Observations:** Characterization of core resilience in hypergraphs.
- **Proposed Method: COREA** - a **fast, effective,** and **theoretically sound** method in improving core resilience via hyperedge addition.
- **Extensive Experiments:**
  - **Superiority: COREA** performs **consistently better** than four competitors on ten real-world hypergraphs in core resilience improvement.
  - **Usefulness**: **COREA** is **useful** for two applications: (a) anomaly detection and (b) identification of influential nodes.

Preliminaries & Problem Definition

(a) Preliminaries:

- **Group interactions:** are common in practice. For example, co-authors of a research paper or participants of a discussion topic.
- **Hypergraph:** $\boldsymbol{G = (V, E)}$ consists of a node set $\boldsymbol{V}$ and an hyperedge set $\boldsymbol{E}$
  - Each hyperedge constitutes a group interaction among people/objects.
- **k-Core:** of $\boldsymbol{G}$ is the maximal sub-hypergraph $\boldsymbol{C_k(G)}$ where each node is incident to at least $\boldsymbol{k}$ hyperedges
- **Core Number:** of node $\boldsymbol{v}$ is the maximum $\boldsymbol{k}$ such that $\boldsymbol{v}$ is in the $\boldsymbol{k}$-Core.
- **Core Resilience:** of $\boldsymbol{G}$ is the *Spearman's rank correlation* of the nodes in $\boldsymbol{V}$ in core numbers *before* and *after* some nodes/hyperedges are removed.
- **Deletion Attack:** of $\boldsymbol{G}$ happens when attackers intrude the system storing the hypergraph (e.x: email database) and delete data (nodes/accounts and hyperedges/records).





(b) Problem Definition: Improving the core resilience of a hypergraph:

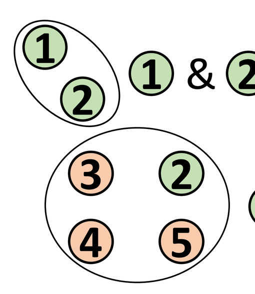
- ***Given***: hypergraph $\boldsymbol{G = (V, E)}$, a budget $\boldsymbol{B} \in \mathbb{N}$
- ***Find***: at most $\mathbf{B}$ hyperedges to augment to $\boldsymbol{G}$
- ***To Maximize***: the core resilience of $\boldsymbol{G}$ against node/hyperedge deletion attack
- ***Subject to Constraints***:
  - All core numbers are preserved
  - The original hyperedge size distribution is conserved

Proposed Concepts & Observations

(a) Proposed Concepts:

- **Anchor(s):** of hyperedge $\boldsymbol{e}$ is/are the nodes having the lowest core number in $\boldsymbol{e}$.
- **Core Strength (CS):** of node $\boldsymbol{v}$ measures the robustness of $\boldsymbol{v}$ in keeping its core number against hyperedge removals.*
- **Core Influence (CI):** of node $\boldsymbol{v}$ measures how $\boldsymbol{v}$ contributes to the core number of its neighbors.*

*Please refer to the paper for exact formulas.

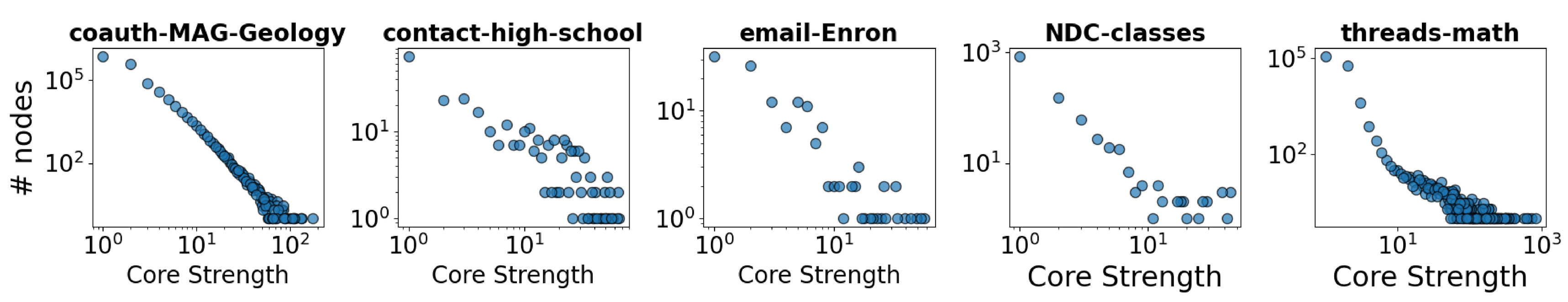


	①	②	③	④	⑤	⑥
CS	1	2	1	1	1	1
CI	5/2	5/2	21/4	21/4	21/4	1

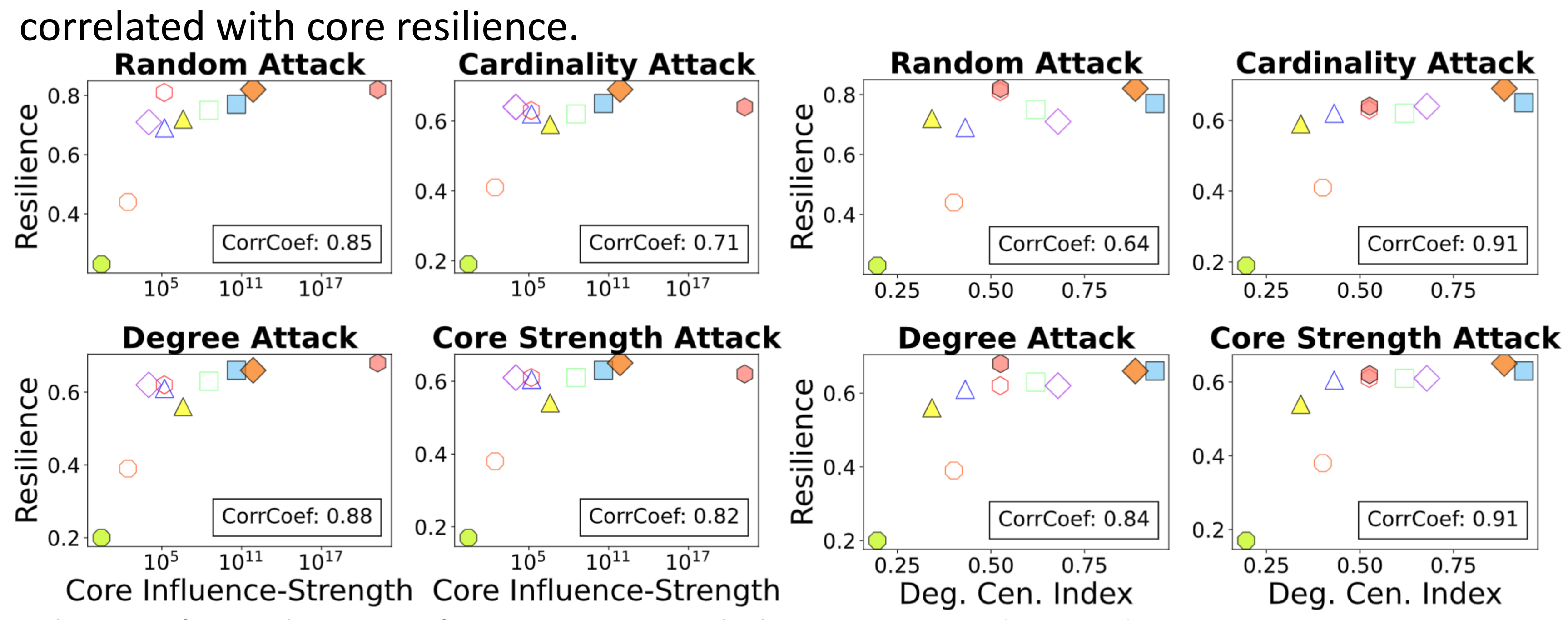
(b) Empirical Observations [1]

- **Datasets:** 10 hypergraphs from 5 domains

Most nodes have low core strengths, leaving room for robustness improvement.



Core influende-strength[2] and Degeneracy Centralized Index[2] are both positively correlated with core resilience.

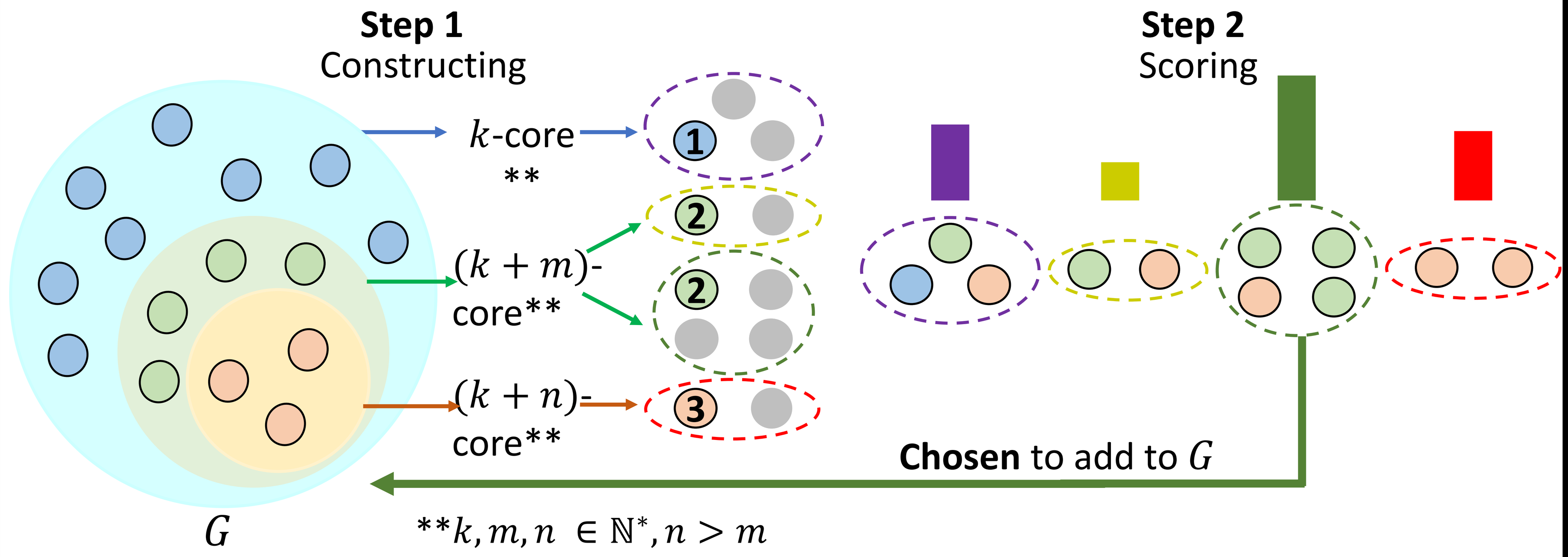


1: Please refer to the paper for more empirical observations and more datasets.
2: Please refer to the paper for the exact formulas.

Proposed Method: COREA

COREA (COre REsilience Improvement Hyperedge Augmentation)

- **Step 1: Candidate Construction:** construct candidate hyperedges that **guarantee to preserve all core numbers.** For each node v of core number k:
  - Step 1-1: determine $c(v)$, the number of hyperedges with v as anchor that can be added to preserve all core numbers.
  - Step 1-2: construct $c(v)$ hyperedges involving v and other nodes from the k-core $C_k(G)$.
- **Step 2. Candidate Selection:** select the **best** candidate hyperedges.
  - Surrogate objective: Core influence-strength of G (correlated with core resilience).
  - Each iteration: choose c candidates of the highest scores to add to G and update the scores of the remaining candidates. Repeat until the budget is exhausted.



Theoretical Merits

[Correctness] COREA returns candidate hyperedges **preserving all core numbers**.

[Invariance] COREA always returns the **same number** of candidate hyperedges.

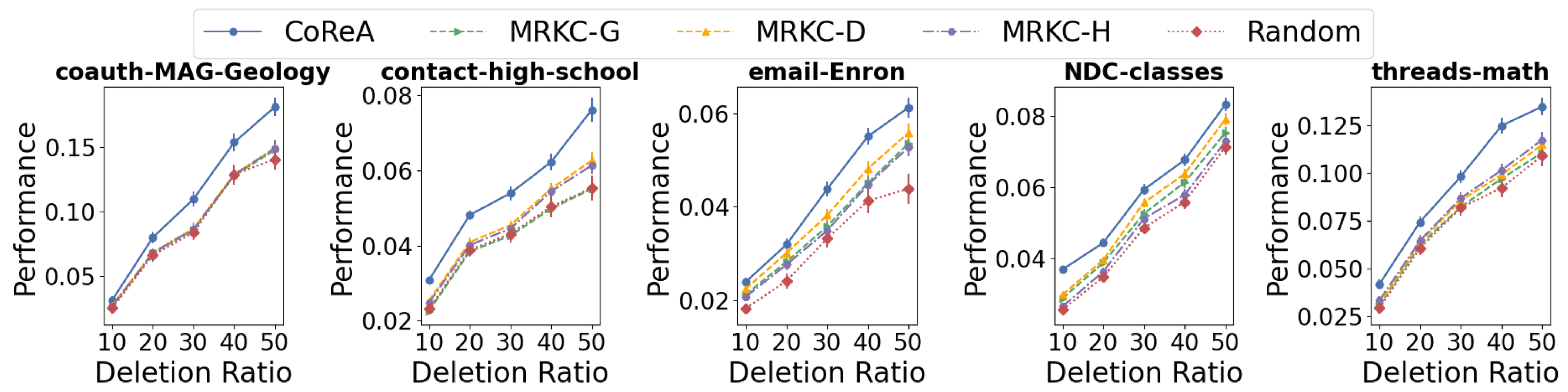
[Exhaustiveness] COREA returns the **maximum number** of candidate hyperedges.

Experiments

- **COREA Improvementation:** Java8
- **Code and Datasets** *:
- **Competitors:** extensions of *MRKC* [1], a graph-based method, to hypergraphs.

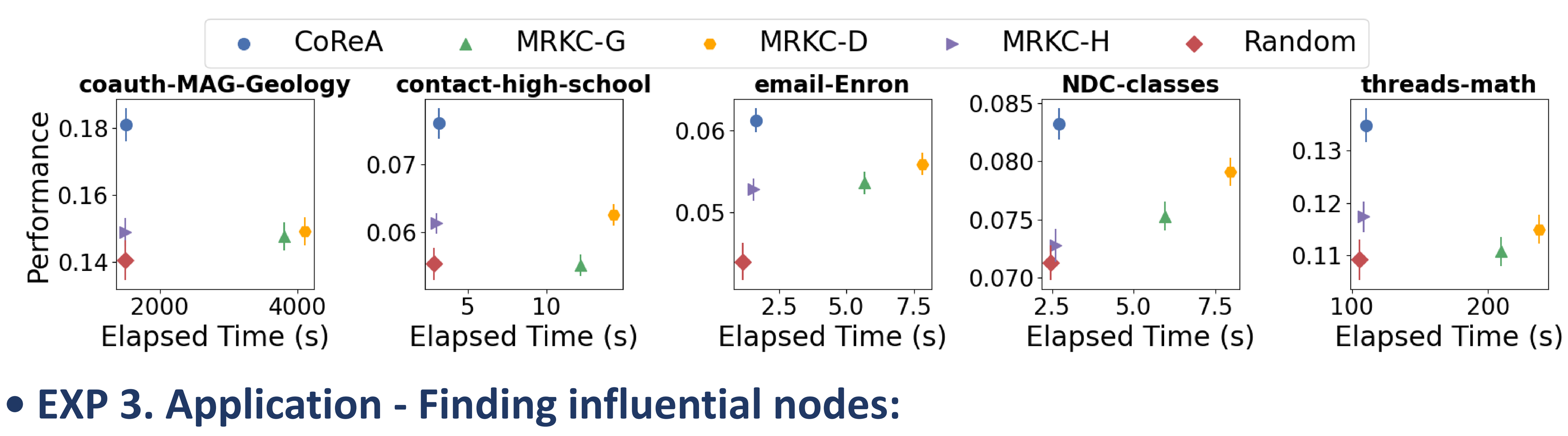
EXP 1. Performance:

We compare the methods in core resilience improvement. Budget: $5\% * |E|$.
COREA consistently outperforms the competitors in core resilience improvement.



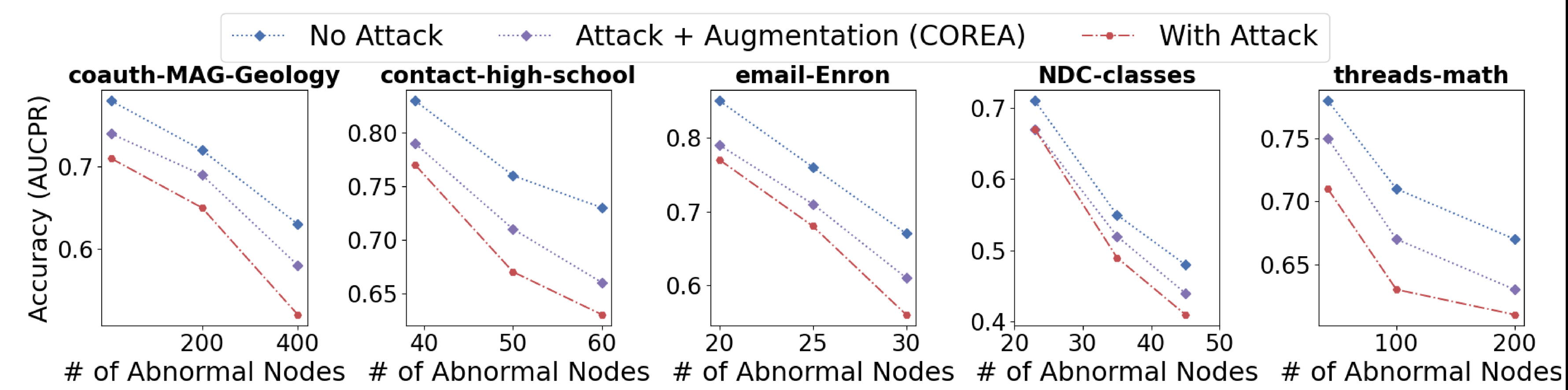
EXP 2. Time-Performance Trade-off:

We compare the running time and performance of the methods. Budget: $5\% * |E|$.
The performance is measured when 50% of the hyperedges are deleted.
COREA consistently provides the best time-performance trade-off.



EXP 3. Application - Finding influential nodes:

We employ the core-based scoring method in [2] to detect abnormal nodes.
We measure the accuracy of the method in the original network, *"No Attack"*, after the hyperedge deletions with the augmentation by **COREA**, *"Attack + Augmentation (COREA)"*, and without such augmentation, *"With Attack"*.
After deletion attack, the core-based method is less useful in predicting anomalies, but the augmentation by **COREA** helps mitigate such decline in usefulness.



* Please refer to the paper for the full results on 10 datasets.

[1] Ricky Laishram et al., *"Measuring and Improving the Core Resilience of Networks"*, WWW 2018.
[2] Kijung Shin et al., *"Patterns and Anomalies in k-Cores of Real-World Graphs with Applications"*, KAIS 2018.