Generating Robust Counterfactual Witnesses for Graph Neural Networks

Dazhuo, Qiu[*], **Mengying Wang**[+], Arijit Khan[*], Yinghui Wu[+]
[*]Aalborg University (Denmark), [+]Case Western Reserve University (USA)

ICDE ’24, May 13–17, 2024, Utrecht, Netherlands

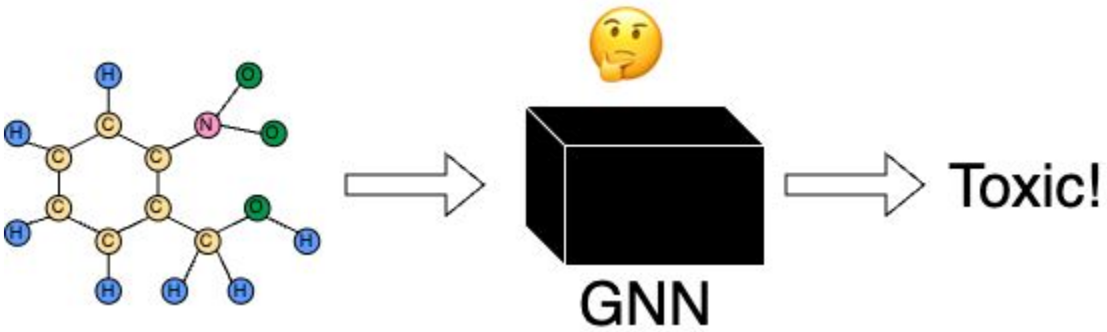
Roadmap

- **Introduction**
  - Background/Motivation
  - Explanation Structures
  - RCW Verification & Generation Problem
- **Methods & Algorithms**
  - A1 - Verification of Witness
  - A2 - Generating Robust Witness
  - A3 - Parallel Witness Generation
- **Experiment**
  - Experiment Settings
  - Experiment Results
- **Conclusion & Future Work**

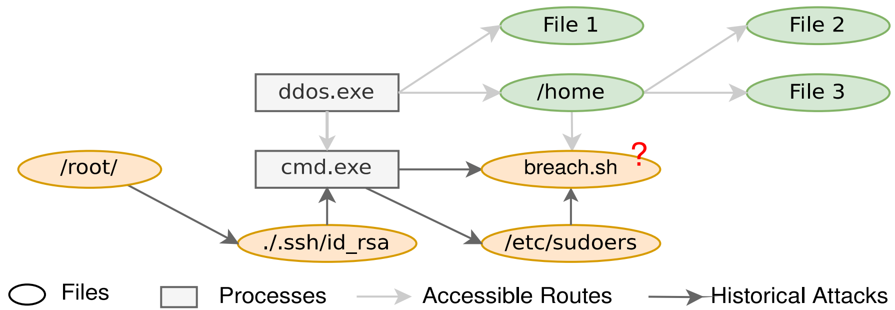
Background/Motivation

- ***"Black-Box" GNNs***:
  - The inference of GNN models are black-box.
  - Hard to understand which part of the input causes the results.

- ***"Explainability"***:
  - Domain experts requires reliable predictions.
  - Highly related to trustworthy challenges.

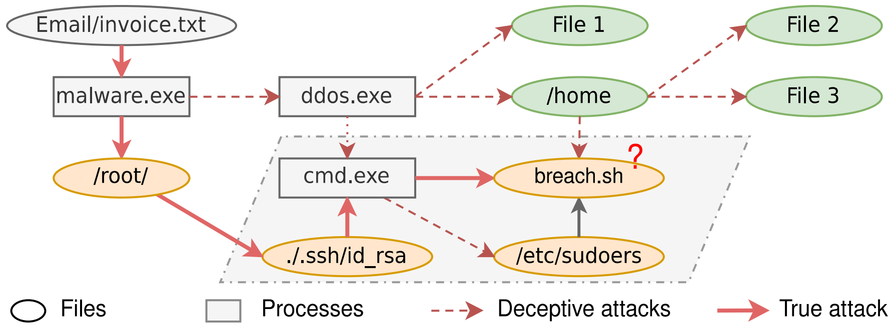


Example - “Vulnerable Zone” in Cyber Networks



🔒 ***GNN-based Security System:***

- **Detection**: Train GNN based on historical attacks to classify files’ vulnerability.
- **Protection**: Enhanced security for vulnerable files (colored orange).



☠️ ***Multi-Phase Cyber Attack Strategy:***

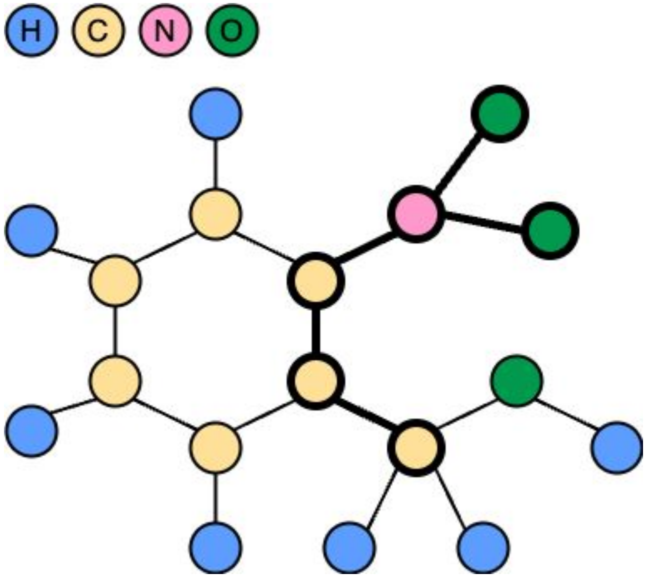
- **Phase 1: Deception Attacks**: Conduct deceptive but harmless attacks to Induce false invulnerable classification on target.
- **Phase 2: True Attack**: attack by exploiting reduced defenses on target.

💡 *How can we identify a "Vulnerable Zone" within cyber networks where, if protected, GNN predictions remain solid, even if other parts of the network are disturbed by deceptive attacks?*

🕐 **Factual Witness,** 🕐**Counterfactual Witness,** 🕐**Robust Counterfactual Witness**

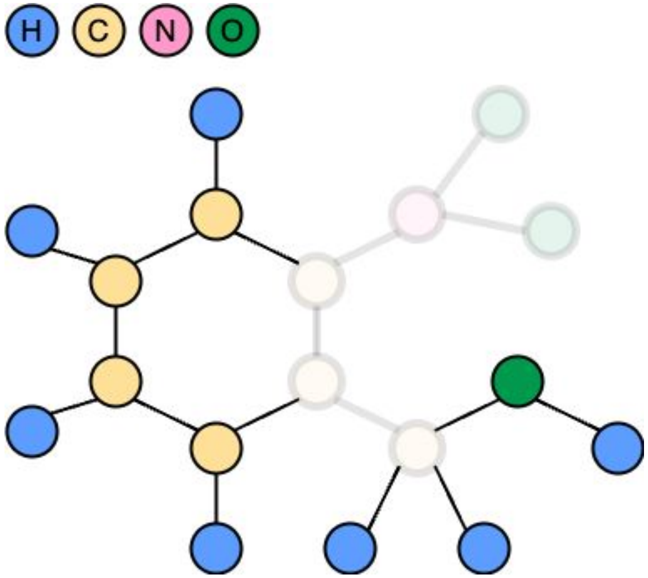
Explanation Structures

- ***Factual Explanation (Witness)***:
  - M(v, G) = M(v, Gs) = l



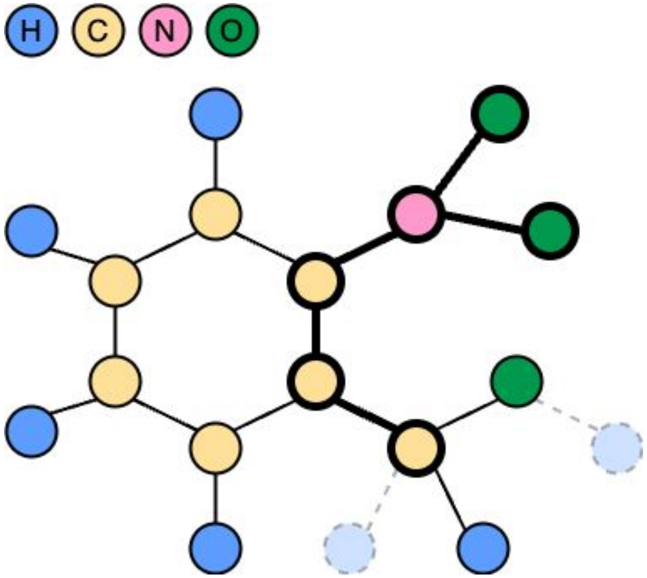
Explanation Structures

- ***Factual Explanation (Witness)***:
  - M(v, G) = M(v, Gs) = l
- ***Counterfactual Explanation (CW)***:
  - M(v, G) ≠ M(v, G\Gs) ≠ l



Explanation Structures

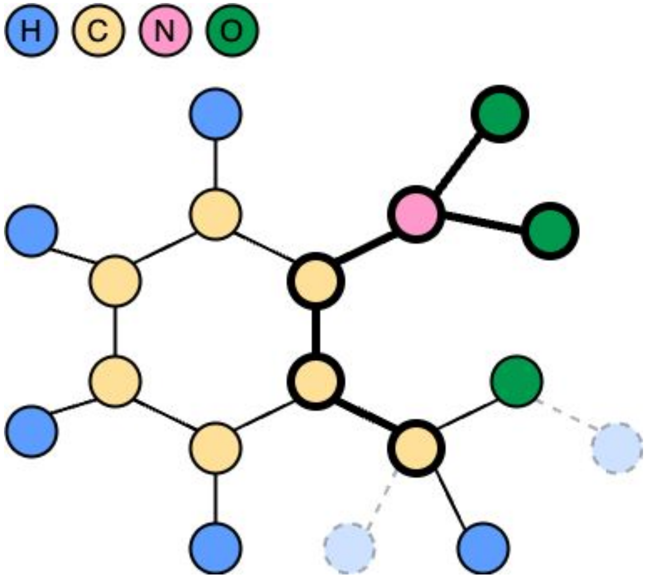
- ***Factual Explanation (Witness)***:
  - M(v, G) = M(v, Gs) = l
- ***Counterfactual Explanation (CW)***:
  - M(v, G) ≠ M(v, G\Gs) ≠ l
- ***Robust Explanation (k-RCW)***:
  - Gs remains consistent under disturbance.



Explanation Structures

- ***Factual Explanation (Witness)***:
  - M(v, G) = M(v, Gs) = l
- ***Counterfactual Explanation (CW)***:
  - M(v, G) ≠ M(v, G\Gs) ≠ l
- ***Robust Explanation (k-RCW)***:
  - Gs remains consistent under disturbance.

We are the first to consider all three criteria! 😎



RCW Verification & Generation Problem

- ***Verification Problem***: Given Gs, decide if Gs is a *k*-RCW for a set of test nodes Vt, w.r.t a model M.
  - Witness verification 👉PTIME.
  - CW verification 👉 PTIME.
  - ***k***-RCW verification 👉NP-hard.

RCW Verification & Generation Problem

- ***Verification Problem***: Given Gs, decide if Gs is a k-RCW for a set of test nodes Vt, w.r.t a model M.
  - Witness verification 👉PTIME.
  - CW verification 👉 PTIME.
  - k-RCW verification 👉NP-hard.

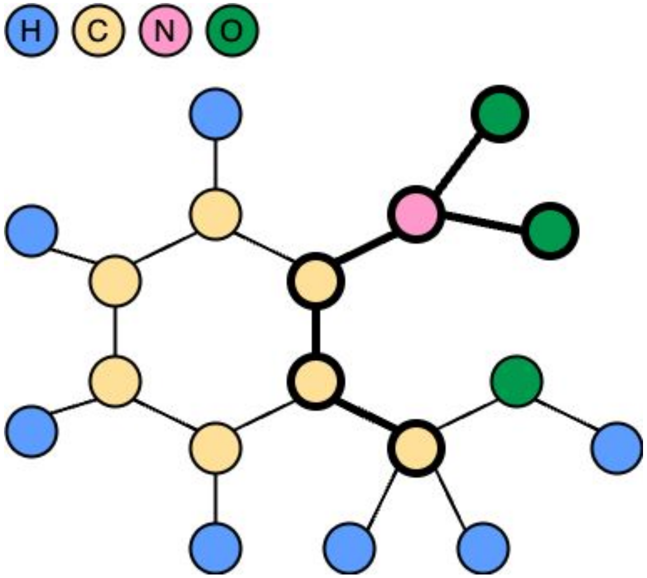
- ***Generation Problem***: Given a graph G and Vt, compute a k-RCW if exists.
  - k-RCW generation in general 👉co-NP-hard
  - under (k, **b**)-disturbances 👉PTIME.

Roadmap

- **Introduction**
  - Background/Motivation
  - Explanation Structures
  - RCW Verification & Generation Problem
- **Methods & Algorithms**
  - A1 - Verification of Witness
  - A2 - Generating Robust Witness
  - A3 - Parallel Witness Generation
- **Experiment**
  - Experiment Settings
  - Experiment Results
- **Conclusion & Future Work**

A1 - Verification of Witness

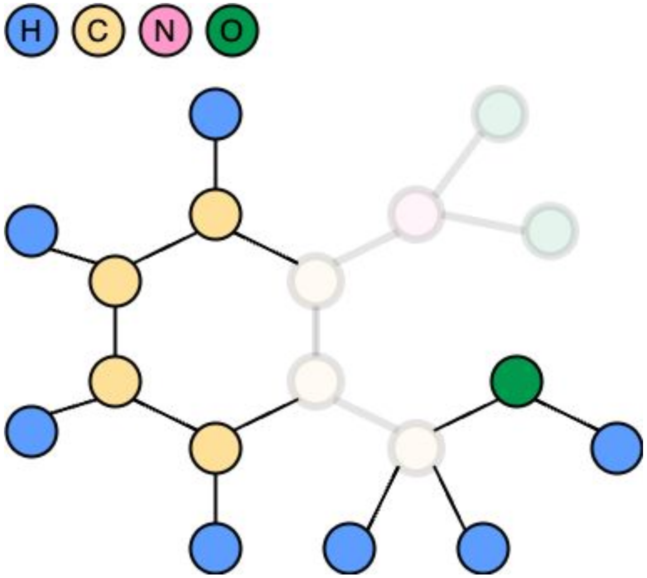
- ***Factual Verification***:
  - Conduct the model inference to verify if the subgraph is a witness.



A1 - Verification of Witness

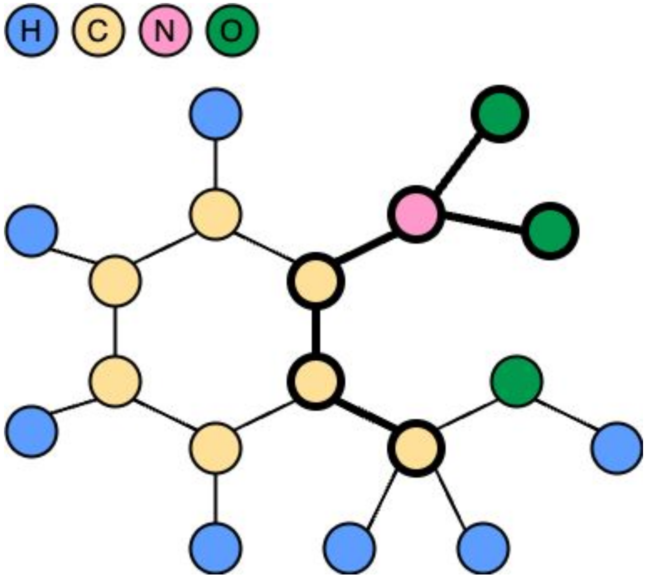
- ***Factual Verification***:
  - Conduct the model inference to verify if the subgraph is a witness.

- ***Counterfactual Verification***:
  - Conduct the model inference to verify if the subgraph is a counterfactual witness.



A1 - Verification of Witness

- ***Factual Verification***:
  - Conduct the model inference to verify if the subgraph is a witness.
- ***Counterfactual Verification***:
  - Conduct the model inference to verify if the subgraph is a counterfactual witness.
- ***Robust Verification***:
  - For each “non-true” label (labels ≠ prediction), verify if the subgraph remains a counterfactual witness under *k* edge flips.



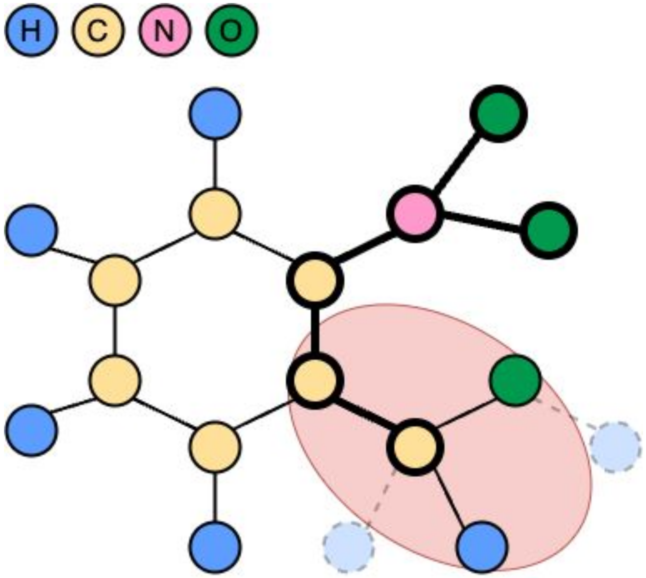
A1 - Verification of Witness

- ***Factual Verification***:
  - Conduct the model inference to verify if the subgraph is a witness.

- ***Counterfactual Verification***:
  - Conduct the model inference to verify if the subgraph is a counterfactual witness.

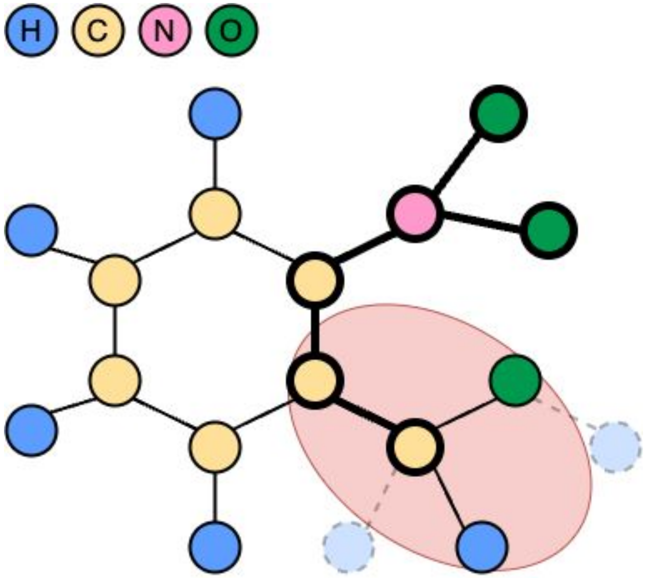
- ***Robust Verification***:
  - For each "non-true" label (labels ≠ prediction), verify if the subgraph remains a counterfactual witness under *k* edge flips.

  - For each node in the "fragile" area (remaining subgraph), select top-*b* edges that are most likely changing the node labels. (PageRank score)



A1 - Verification of Witness

$$O(\underset{\text{\# of classes}}{L}\overset{\text{Size of remaining graph}}{|G'|}(\underbrace{d_m \log d_m}_{\text{Sorting cost of a single node}} + \underbrace{LF(|E| + |V|F)}_{\text{One time APPNP inference cost}}))$$



A2 - Generating Robust Witness

- ***Expand***:

- ***Verify***:

A2 - Generating Robust Witness

- ***Expand***:
  - Includes node pairs that most likely to change its label if “flipped”.
  - Augment the subgraph (initialized with test nodes) with edges that minimize the worst-case margin.

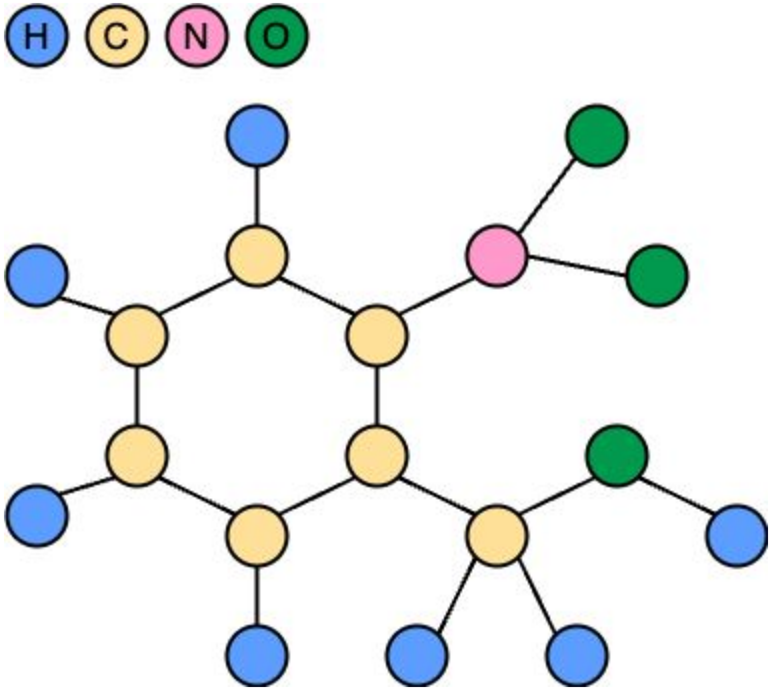
- ***Verify***:

A2 - Generating Robust Witness

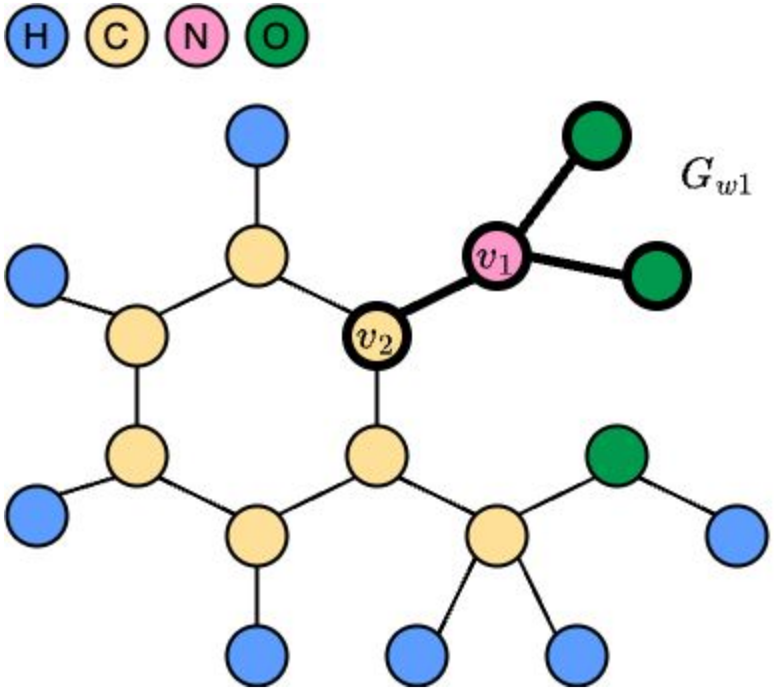
- ***Expand***:
  - Includes node pairs that most likely to change its label if “flipped”.
  - Augment the subgraph (initialized with test nodes) with edges that minimize the worst-case margin.

- ***Verify***:
  - Check if the expanded subgraph is RCW
  - Under k-disturbance: k edges that are most likely to change the prediction.

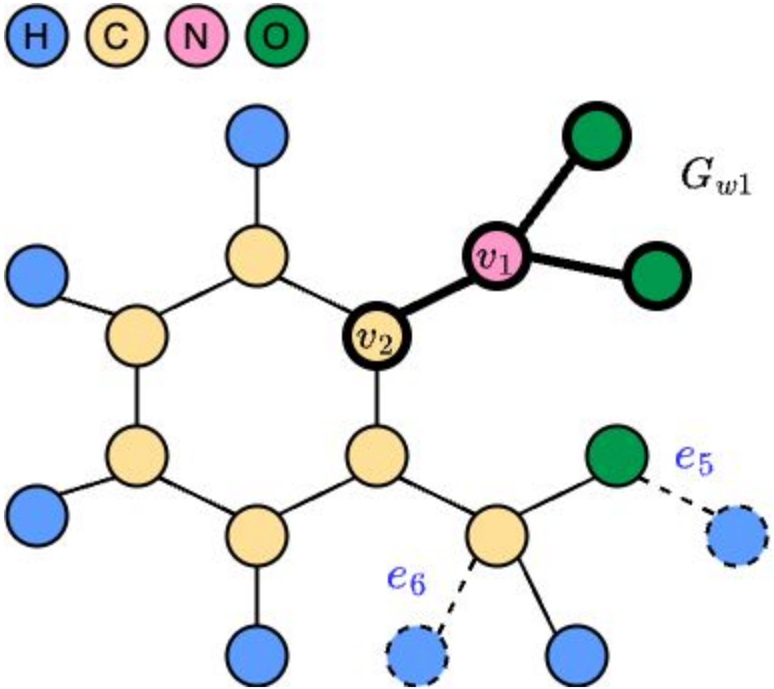
A2 - Generating Robust Witness



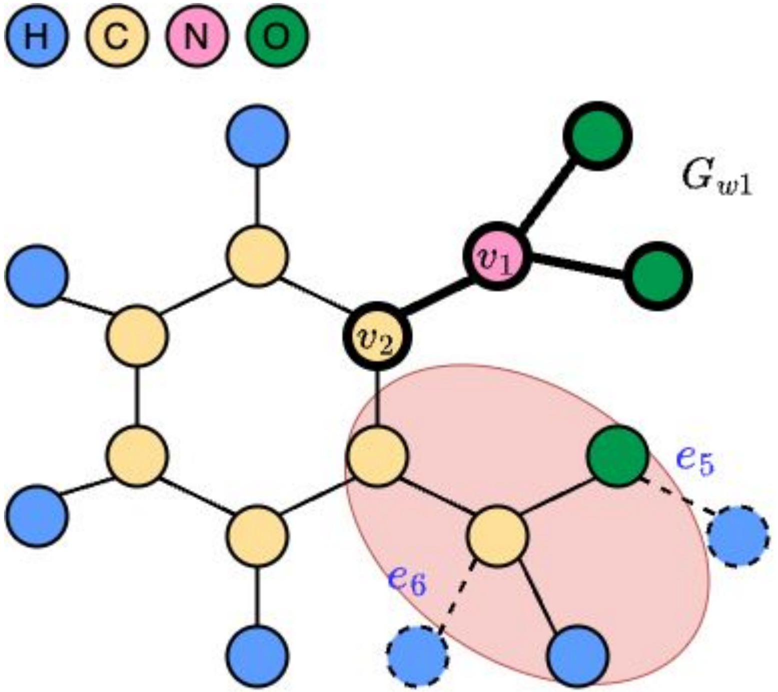
A2 - Generating Robust Witness



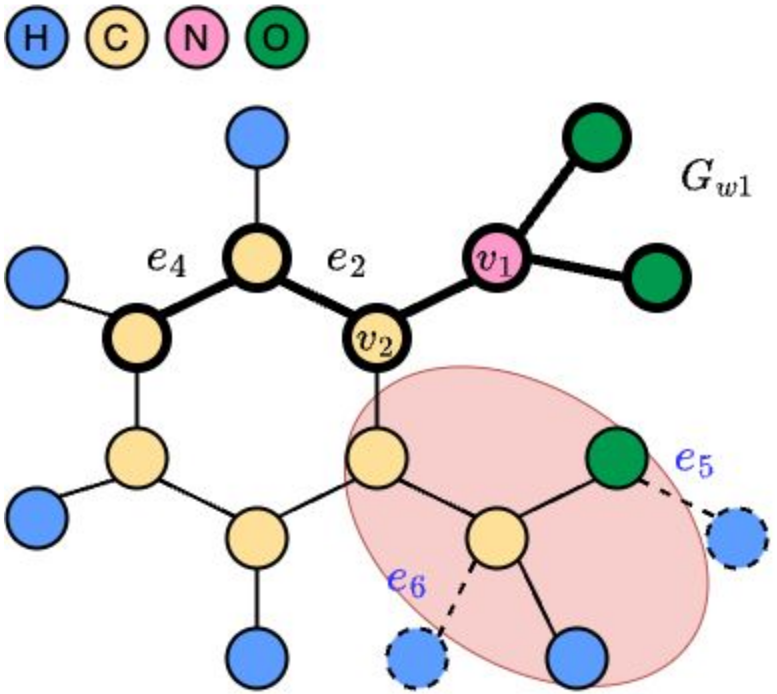
A2 - Generating Robust Witness



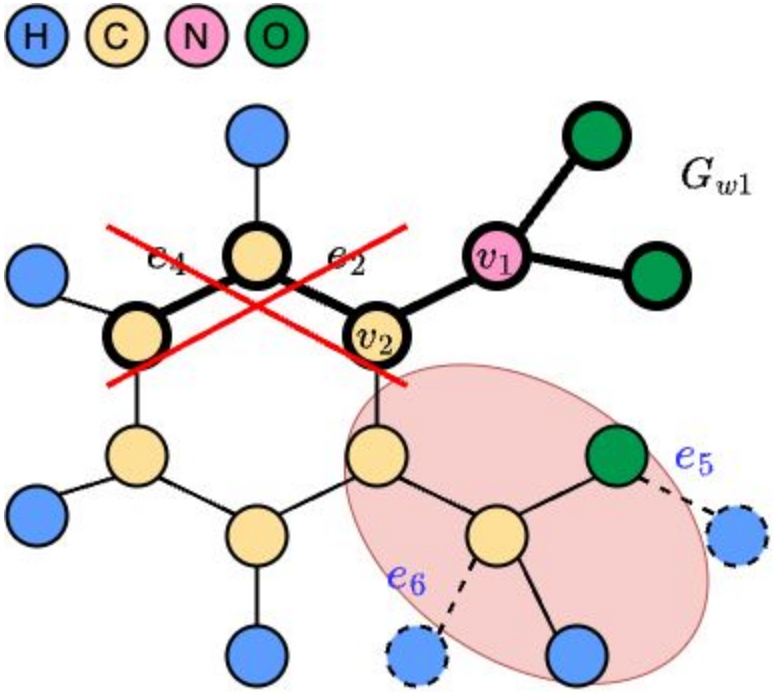
A2 - Generating Robust Witness



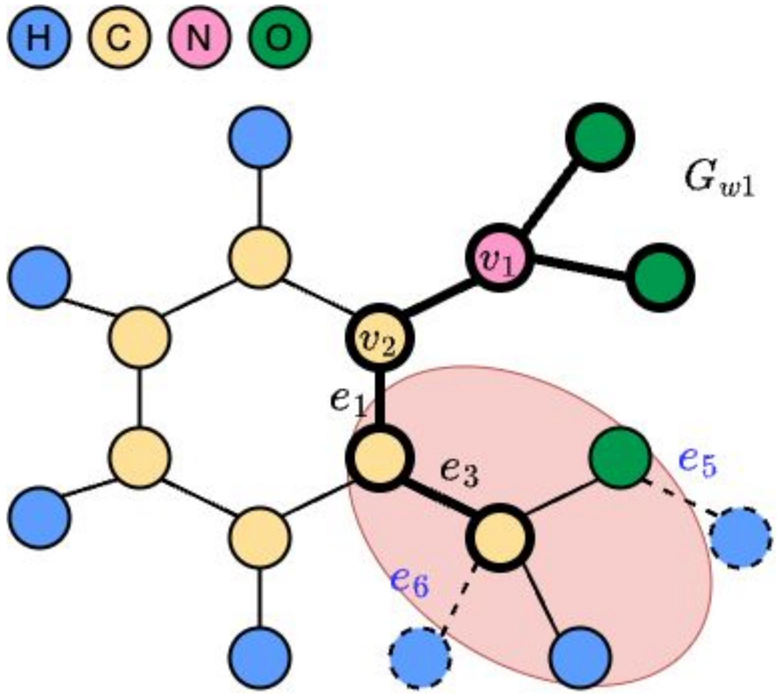
A2 - Generating Robust Witness



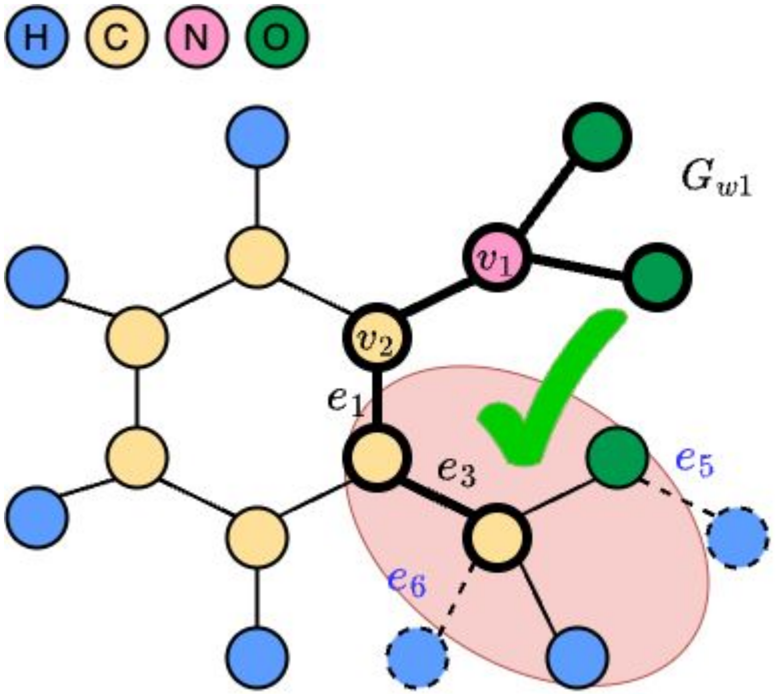
A2 - Generating Robust Witness



A2 - Generating Robust Witness



A2 - Generating Robust Witness



A3 - Parallel Witness Generation

- ***Partition***:
  - Edge-cut based partition where each worker processes one fragment graph.
  - Using a bitmap to record the verified k-disturbance to avoid redundant verification.

- ***Union***:
  - Assemble a global subgraph from each worker with the local subgraph.
  - In each worker expand and verify local subgraph, and maintain the local bitmap.

Roadmap

- **Introduction**
  - Background/Motivation
  - Explanation Structures
  - RCW Verification & Generation Problem
- **Methods & Algorithms**
  - A1 - Verification of Witness
  - A2 - Generating Robust Witness
  - A3 - Parallel Witness Generation
- **Experiment**
  - Experiment Settings
  - Experiment Results
- **Conclusion & Future Work**

Experiment Settings: Datasets

Dataset	# nodes	# edges	# node features	# class labels
BAHouse	300	1500	-	4
PPI	2,245	61,318	50	121
CiteSeer	3,327	9,104	3,703	6
Reddit	232,965	114,615,892	602	41

- ***Different domains.***
  - **BAHouse**: Synthetic.
  - **PPI**: Protein-Protein Interaction.
  - **CiteSeer**: Citation Network.
  - **Reddit**: Social Network.

- ***Large Scale.***
  - **Reddit**: Over one hundred million edges.

Experiment Settings: Baselines

Baselines	Counterfactual	Factual	Robustness
CF-GNNExp (AISTATS 2022)	✓		
CF^2 (WWW 2022)	✓	✓	
RoboGExp	✓	✓	✓

- ***CF-GNNExplainer***:
  - Explainer that considers counterfactual property.
- ***CF^2***:
  - Explainer that considers both counterfactual and factual properties.
- ***RoboGExp***:
  - Our explainer that considers all the properties: counterfactual, factual, and robustness.

Experiment Settings: Evaluation Metrics

- ***Normalized GED***: **(Consistency)**

$$\text{normalized } \mathsf{GED}(G_w, G'_w) = \frac{\mathsf{GED}(G_w, G'_w)}{\max(|G_w|, |G'_w|)}$$

- ***Fidelity+***: **(Counterfactual)**

$$Fidelity+ = \frac{1}{|V_T|}\sum_{v \in V_T}(\mathbb{1}(M(v,G)=l) - \mathbb{1}(M(v, G \setminus G_s) = l))$$

- ***Fidelity-***: **(Factual)**

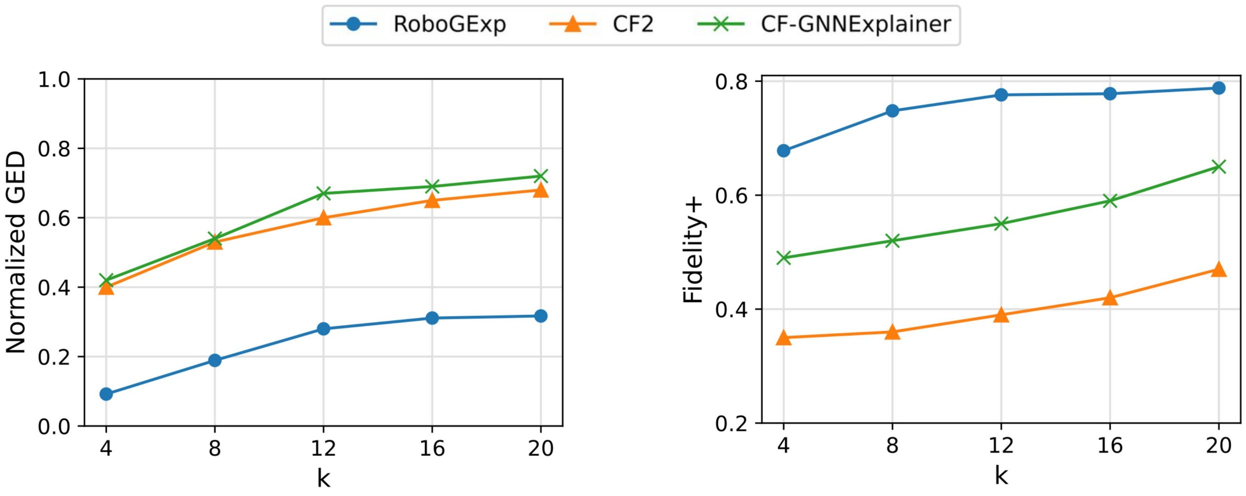
$$Fidelity- = \frac{1}{|V_T|}\sum_{v \in V_T}(\mathbb{1}(M(v,G)=l) - \mathbb{1}(M(v, G_s) = l))$$

Experiment Results: Quality of Explanations

	NormGED	Fidelity+	Fidelity-	Size
RoboGExp	0.32	0.79	0.05	66
CF^2	0.68	0.47	0.06	132
CF-GNNExp	0.72	0.65	0.13	78

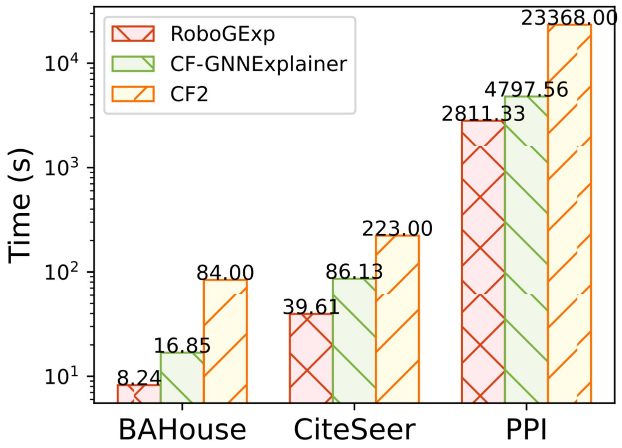
- ***Normalized GED***:
  - Robustness facilitates the consistency of the explanation under disturbance.
- ***Fidelity+ and Fidelity-***:
  - Verification procedure ensures a high fidelity performance.
- ***Size***:
  - RoboGExp integrate the explanation of each test node.

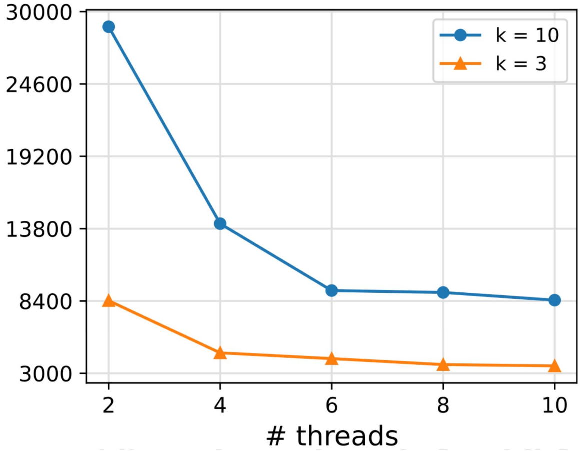
Experiment Results: Effectiveness



- ***Normalized GED (Consistency)***:
    - Outperform baselines even under high disturbance.
- ***Fidelity+ (Counterfactual)***:
    - High disturbance enrich the “fragile” search space.

Experiment Results: Efficiency & Scalability





- ***<u>Generation Time (Efficiency)</u>***:
  - Outperform baselines in various datasets.
- ***<u>Parallel (Scalability)</u>***:
  - Capability of parallelization for scalability.

Roadmap

- **Introduction**
  - Background/Motivation
  - Explanation Structures
  - RCW Verification & Generation Problem
- **Methods & Algorithms**
  - A1 - Verification of Witness
  - A2 - Generating Robust Witness
  - A3 - Parallel Witness Generation
- **Experiment**
  - Experiment Settings
  - Experiment Results
- **Conclusion & Future Work**

Conclusion & Future Work

- ***Conclusion***:
  - Explanation structure: *k*-robust counterfactual witness (*k*-RCW).
  - Feasible algorithms for verification and generation problems with impressive results.

- ***Future Work***:
  - Minimal/Minimum explanations.
  - Extension to other GNN-based applications.

THANK YOU !

Email: dazhuoq@cs.aau.dk, mxw767@case.edu

GitHub: https://github.com/DazhuoQ/RoboGExp

Acknowledgment: Qiu and Khan are support from the Novo Nordisk Foundation grant NNF22OC0072415. Wang and Wu are supported in part by NSF under CNS-1932574, ECCS-1933279, CNS-2028748 and OAC-2104007.