Tutorial: GNN Explainers 2.0: User-centric and Data Driven Insights



Arijit Khan, Xiangyu Ke, Yinghui Wu, Francesco Bonchi

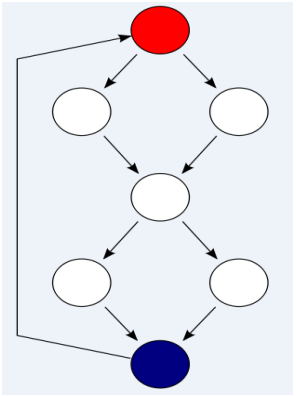
Introduction

Arijit Khan

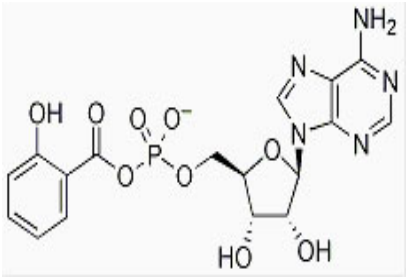
Graph data is everywhere

Graph database with many smaller graphs

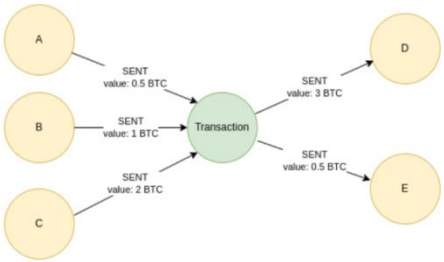
One large graph



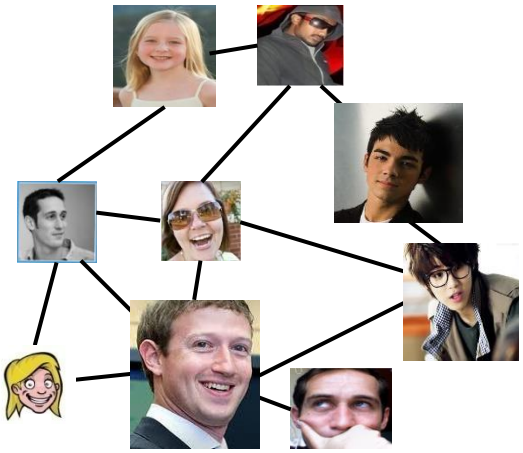
Program flow/ call graph



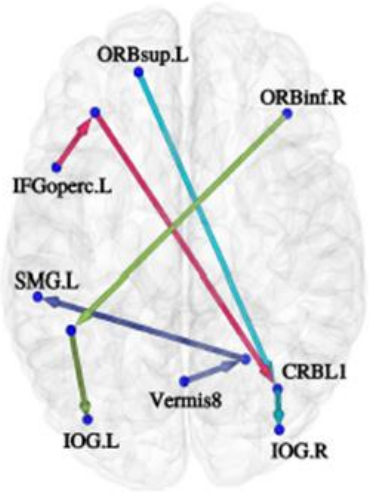
Chemical compound structure

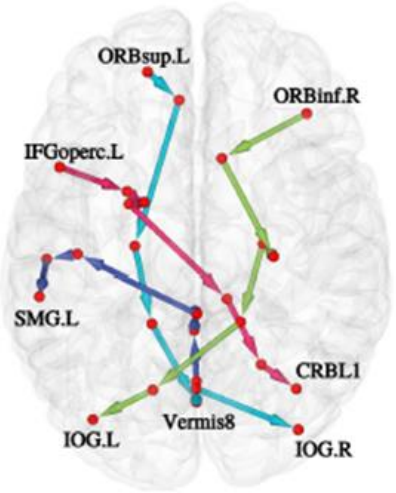


Financial transaction

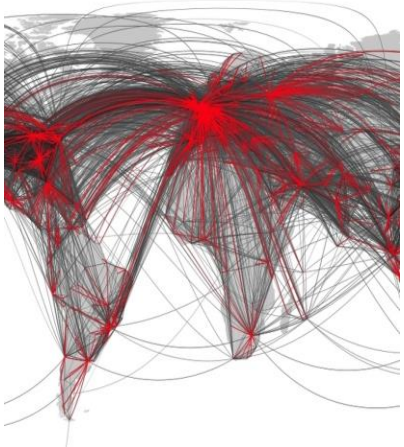


Social network

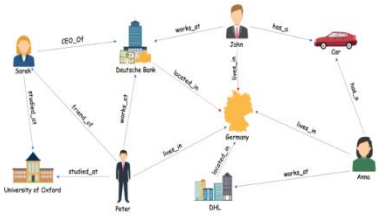




Human brain network



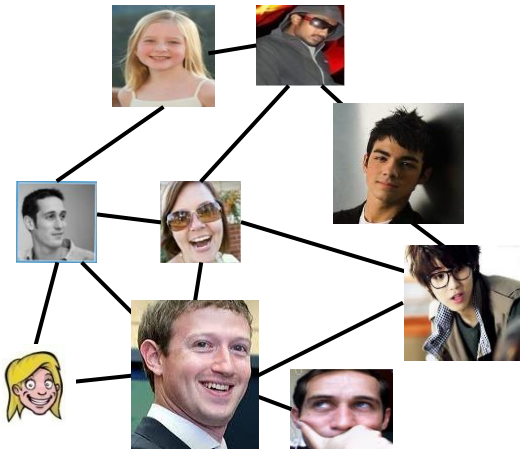
Transportation



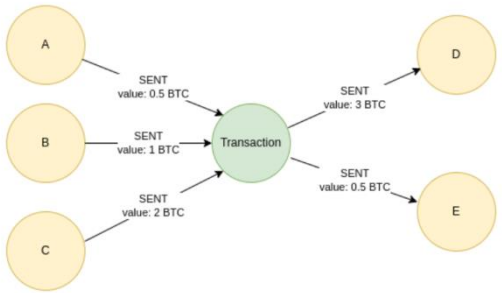
Knowledge graph

Graph data is everywhere

One large graph

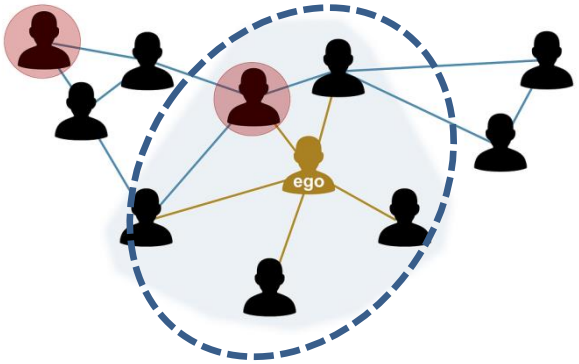


Social network

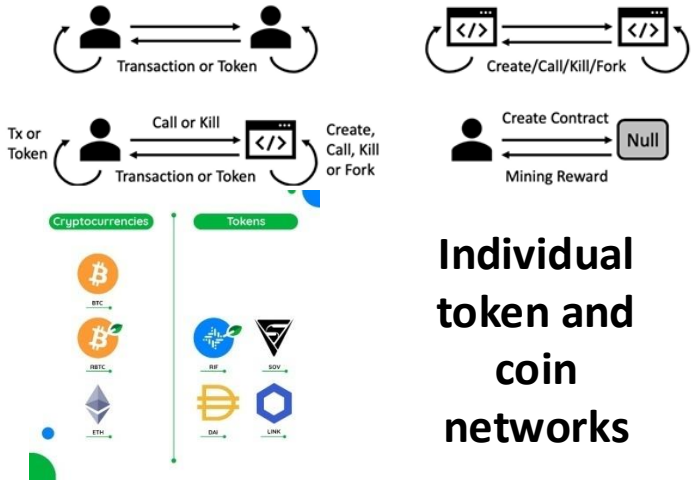


Financial transaction

Graph database with many smaller graphs

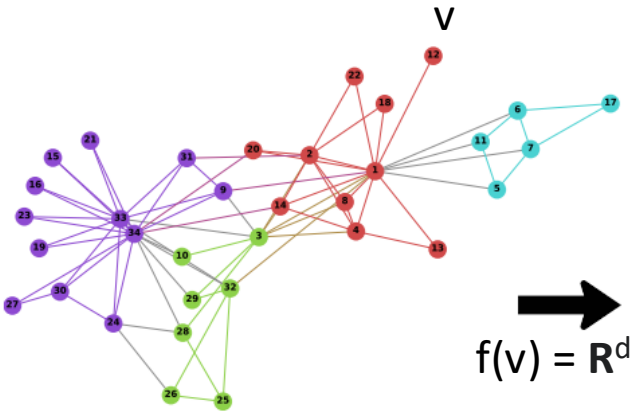


Ego network of individual users

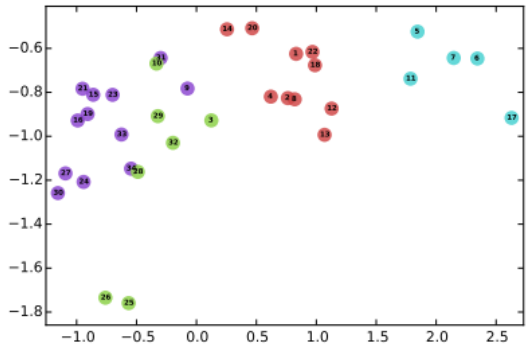


Individual token and coin networks

Graph neural network (GNN): Key idea



$f(v) = \mathbf{R}^d$



Representation learning or embedding

Downstream tasks (e.g., graph and node classification, recommendation, link prediction, question answering)

Learning could be end-to-end

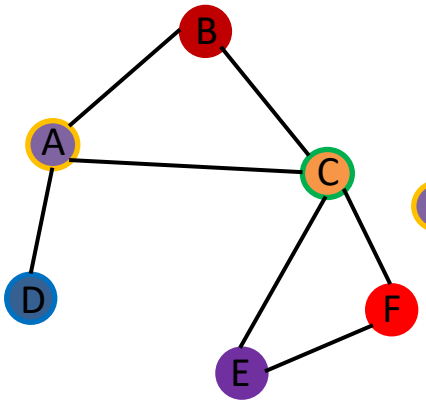
- End-to-end learning → ~~Feature Engineering~~
- Task-independent / task-dependent learning.
- Can capture graph structure and node, edge features.

Graph Convolutional Neural Network (GCN)

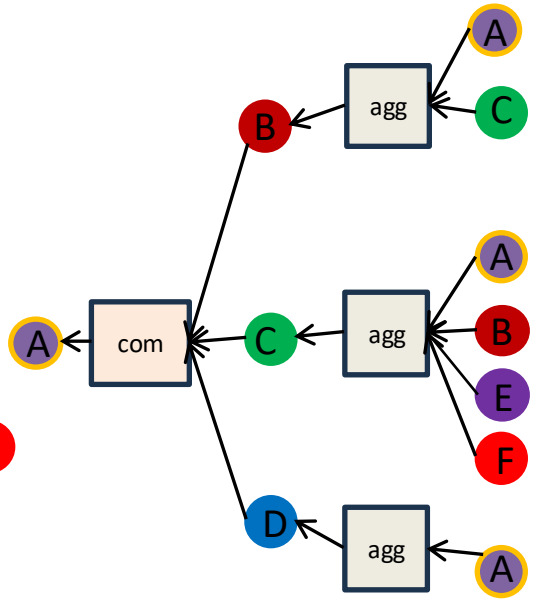
- Massage passing to use aggregation and combine functions repeated several times.

$$H^{(t+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}.\tilde{A}.\tilde{D}^{-\frac{1}{2}}.H^{(t)}.W^{(t)})$$

$\tilde{A} = A + I_N$ $\quad H^{(0)} = X$ Input Node Features



Input graph



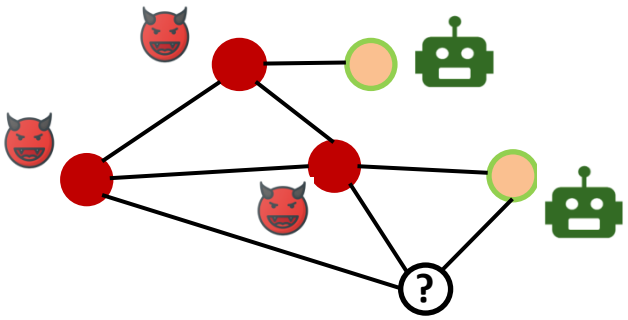
Technique in GCN

Representation Learning on Networks (WWW Tutorial, 2018)

Graph neural network (GNN): Downstream tasks

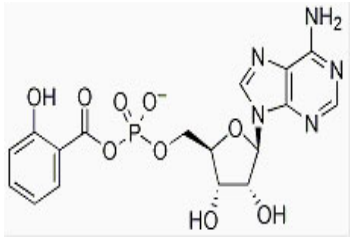
Node classification

Node-level task



Graph classification

Graph-level task

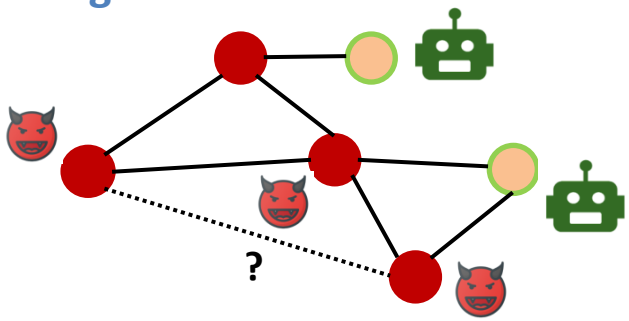


? Mutagen

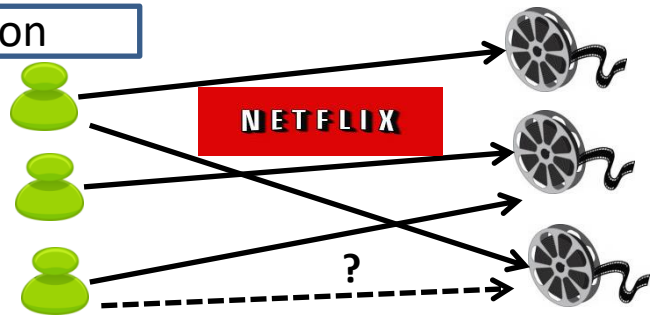
? Non-mutagen

Link prediction

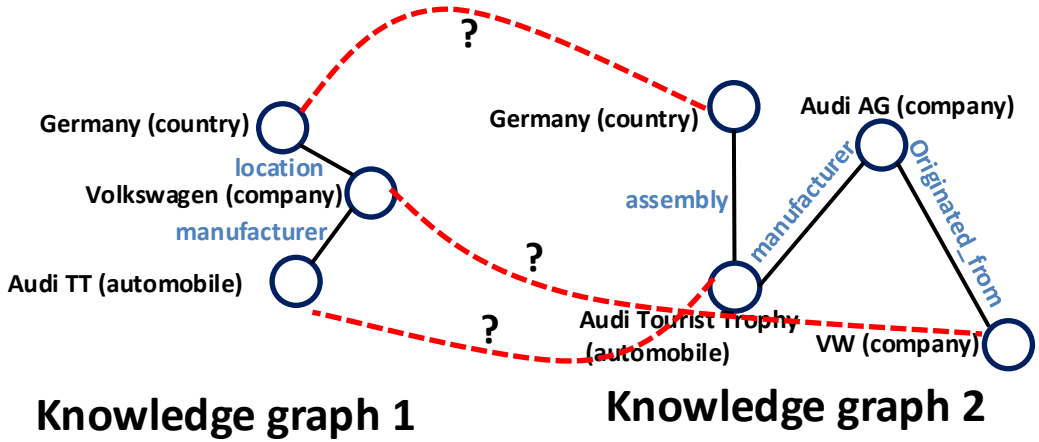
Edge-level task



Recommendation



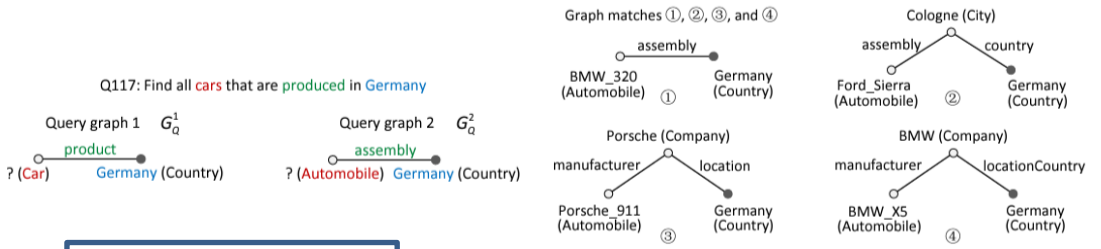
Entity resolution



Knowledge graph 1

Knowledge graph 2

Question answering



Drug design

- predicting missing links between drug and disease.

Graph Neural Network (GNN): Explainability [IEEE DSAA 2023 Tutorial]

- Explain the results of high-quality GNNs.
- **[Instance-level]** Understand which aspects of the input data drive the decisions of the GNN – discover critical nodes, edges, subgraphs, and their features that are responsible for GNN outcomes.
- **[Model-level]** Insight on how GNNs work – discover what input subgraph patterns lead to a certain prediction.

Importance

- Desirable to understand and explain the workings and results of black-box GNNs – bridge domain knowledge with GNN predictions, human-AI collaboration.
- Safety and well-being (e.g., autonomous car, AI in healthcare) – trust in deep learning models.
- Understand bias in machine learning (ML) algorithms – ML algorithms can amplify bias, model debugging.
- Robustness against adversarial examples – improve quality of GNN outputs.
- Legal requirements, e.g., GDPR – algorithms to explain their outputs.
- Scientific applications – identify hidden patterns, rules, root causes, discover laws in science (biology, physics, chemistry, social science, etc.)

Stakeholders

End users, domain experts, decision makers, policy makers, regulatory agencies, researchers, data scientists, and engineers

Challenges with GNN Explanations

- Many definitions, motivations, and requirements for explainability.

– trust, causality, transferability, informativeness, fair and ethical decision making, model debugging, recourse, mental model comparison, context-dependent, low-level mechanistic understanding of models, high-level human understanding, what makes users confident about the model.

- Comparing explanations is hard!

- Several quantitative and qualitative evaluation metrics or methods.

– **Quantitative:** faithfulness (fidelity+, fidelity-), sparsity, contrastivity, accuracy, stability.

– **Qualitative:** application-grounded, human-grounded, and functionally-grounded evaluation.

- Difficult to obtain ground-truth.

- Security and privacy concerns.

- **Other issues:** Evaluation via occlusion creates data outside training distribution, bias terms, redundant evidence, trivial correct explanations, weak GNN model, misaligned GNN architecture, problems due to graph data vs. grid data.

- Capture interplay of graph structure and features in GNN's decision making.

Challenges with GNN Explanations

- Many definitions, motivations, and requirements for explainability

– trust, causality, transferability, informativeness, fair and ethical decision ma
mental model comparison, context-dependent, low-level mechanistic u
understanding, what makes users confident about the model.

- Comparing explanations is hard!

- Several quantitative and qu

– **Quantitative:** faithfulne

– **Qualitative:**

- Diff

- Securi

- **Other issu** occlusion creates data outside training distribution, bias terms, redundant evidence, trivia planations, weak GNN model, misaligned GNN architecture, problems due to graph data vs. grid data.

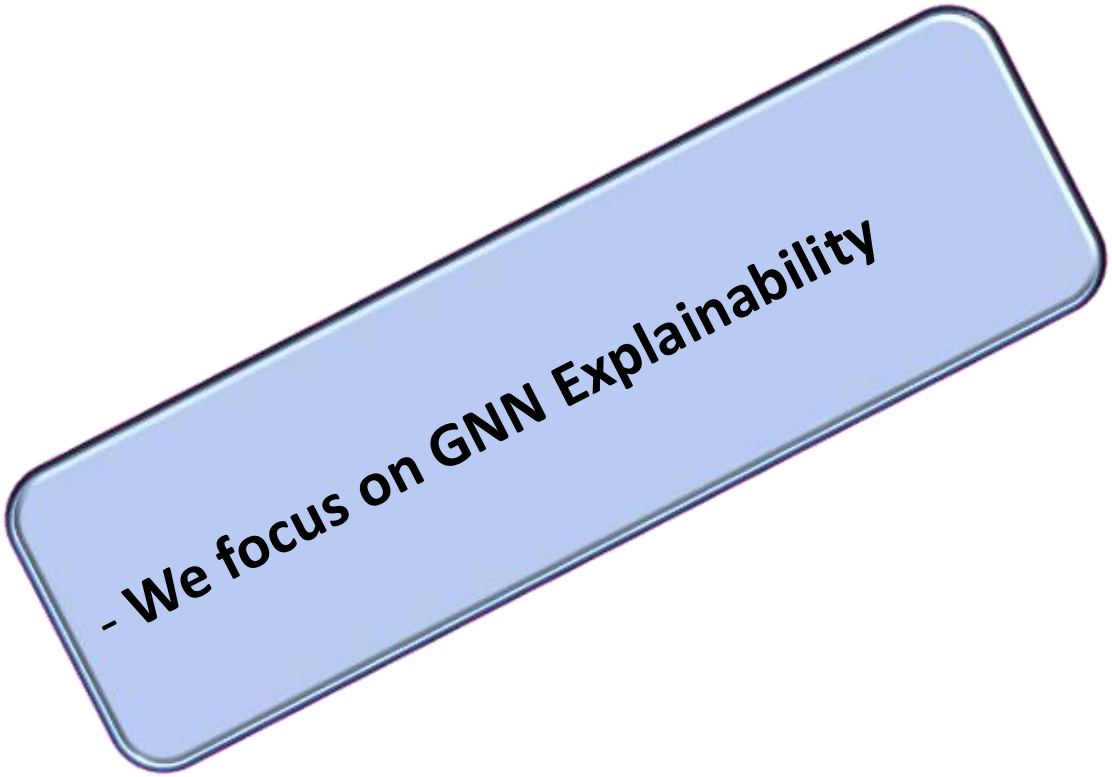
- Capture interplay of graph structure and features in GNN's decision making.

- **USER-FOCUSED EXPLANATION:** "Ability to explain or to present a model in understandable terms to humans"
- Doshi-Velez and Kim 2017

Interpretability vs. Explainability

• Often used interchangeably.

• Interpretability concerns the understanding (of inner workings) of the model by AI experts and researchers, while explainability focuses on explaining the decisions made to end users.



Tutorial outline

1 Introduction

1.1 Graph neural networks (GNNs) and applications

1.2 Explainability of GNNs

- Definitions, importance, and challenges

2 GNN Explainers Categorization

3 GNN Explainers 2.0

4 User-centric and Data-driven Explainability Methods for GNNs

5 Future directions

Tutorial outline

1 Introduction

2 GNN Explainers Categorization

- 2.1 Post-hoc vs. intrinsic / self-explainable
- 2.2 Global vs. local
- 2.3 Class-specific vs. instance-specific
- 2.4 Model-specific vs. model-agnostic
- 2.5 Forward vs. backward
- 2.6 Node-level vs. edge-level vs. subgraph-level
- 2.7 Perturbation vs. gradient vs. decomposition vs. surrogate models
- 2.8 Factuals vs. counterfactuals

3 GNN Explainers 2.0

4 User-centric and Data-driven Explainability Methods for GNNs

5 Future directions

GNN Explainers Categorization

Arijit Khan

Explainability in Graph Neural Networks: A Taxonomic Survey.
Hao Yuan, Haiyang Yu, Shurui Gui, Shuiwang Ji. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 45, Issue 5, Pages 5782 - 5799, 2023.

Jaykumar Kakkad, Jaspal Jannu, Kartik Sharma, Charu Aggarwal, Sourav Medya.
A Survey on Explainability of Graph Neural Networks. IEEE Data Eng. Bull. 47(2): 35-63 (2023)

Post-hoc vs. Intrinsic Explainability Methods

We'll focus on them

• **Post-hoc.** Creating a second model/ algorithm to provide explanations for an existing GNN model.

- e.g., perturbation-based approaches (GNNExplainer, NeurIPS2019).

- could be limited in explanation performance (e.g., reporting spuriously-correlated features with the task), while keeping the underlying GNN accuracy intact.

• **Intrinsic / self-explainable.** Constructing self-explanatory models which incorporate explainability directly to their structures.

- e.g., use structural constraints to derive an informative subgraph which is used for both prediction and explanation.

- e.g., graph attention networks (ICLR 2018), SEGNN (CIKM 2021), ProtGNN (AAAI 2022).

- trade-off between good explainability vs. prediction accuracy.

Global vs. Local Explainability Methods

• **Global.** Explain the *overall behavior* of the model, e.g., XGNN (KDD 2020), PGExplainer (NeurIPS 2020), PGM-Explainer (NeurIPS 2020), GraphMask (ICLR 2021).

- Provide model-level summaries.
- Capture common structural motifs, feature importance trends, or decision rules
- Often approximate the model with a simpler surrogate (e.g., rule sets, prototypes)

• **Local.** Explain the *reasoning behind a specific prediction,* e.g., GNNExplainer (NeurIPS 2019), SubgraphX (ICML 2021), CF-GNNExplainer (AISTATS 2022).

- Provide instance-level explanations.
- Highlight subgraphs, features, or neighbors that influenced one prediction.
- Often produce masks, saliency maps, or counterfactuals.

Class-specific vs. Instance-specific Explainability Methods

• **Class-specific.** Explain what the model considers important for predicting a *particular class,* e.g., XGNN (KDD 2020).

- Aggregate explanations across many instances of the same class.
- Produce class-level prototypes, motifs, or feature sets.
- Useful for fairness, consistency, and debugging class boundaries.

• **Instance-specific.** Explain the prediction for *one specific instance.,* e.g., GNNExplainer (NeurIPS 2019), SubgraphX (ICML 2021), CF-GNNExplainer (AISTATS 2022).

- Provide explanations tailored to a single sample.
- Often produce subgraph masks, feature attributions, or counterfactuals.
- Sensitive to local neighborhood and feature perturbations.

Model-specific vs. Model-agnostic Explainability Methods

• **Model-specific / white-box.** Requires access to internal model parameters or embeddings to provide explanations.

- e.g., gradient-based methods calculate the gradient of an output w.r.t. the input using backpropagation to derive the contribution of features (*Explainability methods for graph convolutional neural networks*. in CVPR, 2019).
- Gradient-based methods are more efficient, since they usually need one forward and another backward pass.
- **Issues:** gradient saturation, ..

• **Model-agnostic / black-box.** Does not require internals of the GNNs to generate explanations.

- e.g., perturbation-based methods (GNNExplainer, NeurIPS 2019) determine the contribution of a feature by measuring how prediction score changes when the feature is altered.
- can be computationally inefficient as each perturbation requires a separate forward propagation through the network.

Forward vs. Backward Explainability Methods

• **Forward interpretability methods**. GNN model-agnostic, learn evidence about graphs or nodes passed through the GNN.

- e.g., perturbation-based, that is, masking some node features and/ or edge features and analyzing the changes when the modified graphs are passed through the GNN model.
- e.g., employ a simple, interpretable surrogate model to approximate the predictions of a complex GNN.

• **Backward interpretability methods.** GNN model-specific.

- e.g., gradient-based – backpropagating importance signal from the output neuron of the model to the individual nodes of the input graph.

- e.g., decomposition-based – distributing the prediction score in a backpropagation manner until the input layer.

Node-level vs. Edge-level vs. Subgraph-level Explainability Methods

- **Output of explainability methods**. Node / node feature, edge, subgraph.

- Node / node feature. E.g., ZORRO, IEEE Transactions on Knowledge & Data Engineering. 35(8), 2023.

- Edge. E.g., PGExplainer, NeurIPS 2020.

- Subgraph. E.g., SubgraphX, ICML 2021.

Counterfactual vs. Factual Explainability Methods

• **Factual Explanation.** Finding a subgraph whose information is sufficient to lead to the same prediction for the input sample. E.g., GNNExplainer, NeurIPS 2019.

• **Counterfactual Explanation.** Finding a subgraph whose information is necessary hence its removal will result in different predictions (i.e., necessary subgraph for the targeted class). E.g., GCFExplainer, WSDM 2023.

• **Factual and Counterfactual Explanation.** Finding a subgraph that follows both the factual and counterfactual reasoning; finding a subgraph that outputs the same prediction and its absence will cause changes on the output of the model. E.g., CF^2, WWW 2022.

Different Explainability Methods for GNNs (Instance-Specific)

• **Perturbation-based.** Masking some node features and/ or edge features and analyzing the changes when the modified graphs are passed through the GNN model. E.g., GNNExplainer, NeurIPS 2019.

• **Surrogate model.** Employs a simple, interpretable surrogate model to approximate the predictions of a complex GNN. E.g., PGM-explainer, NeurIPS 2020.

• **Gradient-based.** Backpropagating importance signal from the output neuron of the model to the individual nodes of the input graph. E.g., Grad-CAM (*Explainability methods for graph convolutional neural networks*. in CVPR, 2019).

• **Decomposition-based.** Distributing the prediction score in a backpropagation manner until the input layer. E.g., LRP, TextGraphs 2019.

Different Explainability Methods for GNNs (Global)

• **Generation-based.** Graph generator generates example graph patterns that maximize the prediction probability of each class. E.g., XGNN, KDD 2020.

• **Global counterfactuals-based.** Find a small set of representative counterfactual graphs that explains all input graphs. E.g., GCFExplainer, WSDM 2023.

• **Global concept-based.** GNN neurons as global concept detectors. E.g., *Global concept-based interpretability for graph neural networks via neuron analysis*, AAAI 2023.

Tutorial outline

1 Introduction 

2 GNN Explainers Categorization 

3 GNN Explainers 2.0 

4 User-centric and Data-driven Explainability Methods for GNNs

5 Future directions

GNN Explainers 2.0

Arijit Khan

Limitations of SOTA GNN Explainers

GNN Explainers 1.0; e.g., GNNExplainer, PGExplainer, SubgraphX, PGMExplainer, GraphLime, GCFExplainer, CF2, GNN-LRP)

- Explaining model behavior
- One-off explanations of the model's final output
- Less interactive, less user-focused
- Task-limited, e.g., mainly for node and graph classification

User-centric and Data-driven Explanations

GNN Explainers 2.0

- Explanations for end-users and domain experts
- Layer-wise provenance for model debugging and optimization
- Interactive, configurable, efficient, and scalable explanations
- Robust and multi-criteria optimization explanations
- Explanatory query and output interface based on structured query, natural language, and examples

Tutorial outline

1 Introduction

2 GNN Explainers Categorization

3 GNN Explainers 2.0

4 User-centric and Data-driven Explainability Methods for GNNs

4.1 Pattern Mining and Concept Hierarchies
4.2 Model-slicing Explanations
4.3 Robust Explanations
4.4 Multi-criteria Explanations
4.5 Declarative Explanatory Queries
4.6 Efficiency and Interactiveness
4.7 Natural Language Explanations
4.8 Counterfactual Explanations

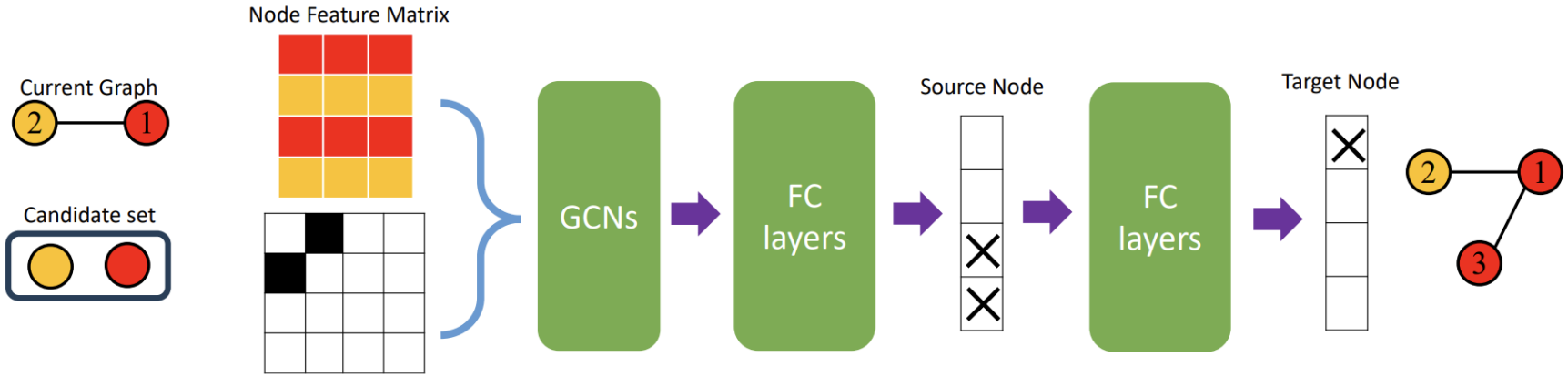
5 Future directions

Pattern Mining and Concept Hierarchies

Xiangyu Ke

Pattern Mining and Concept Hierarchies

Current explanations usually emphasize numerical logits or feature scores over many domain-specific structures, making explanations *hard to interpret, access, and adapt for downstream use*.

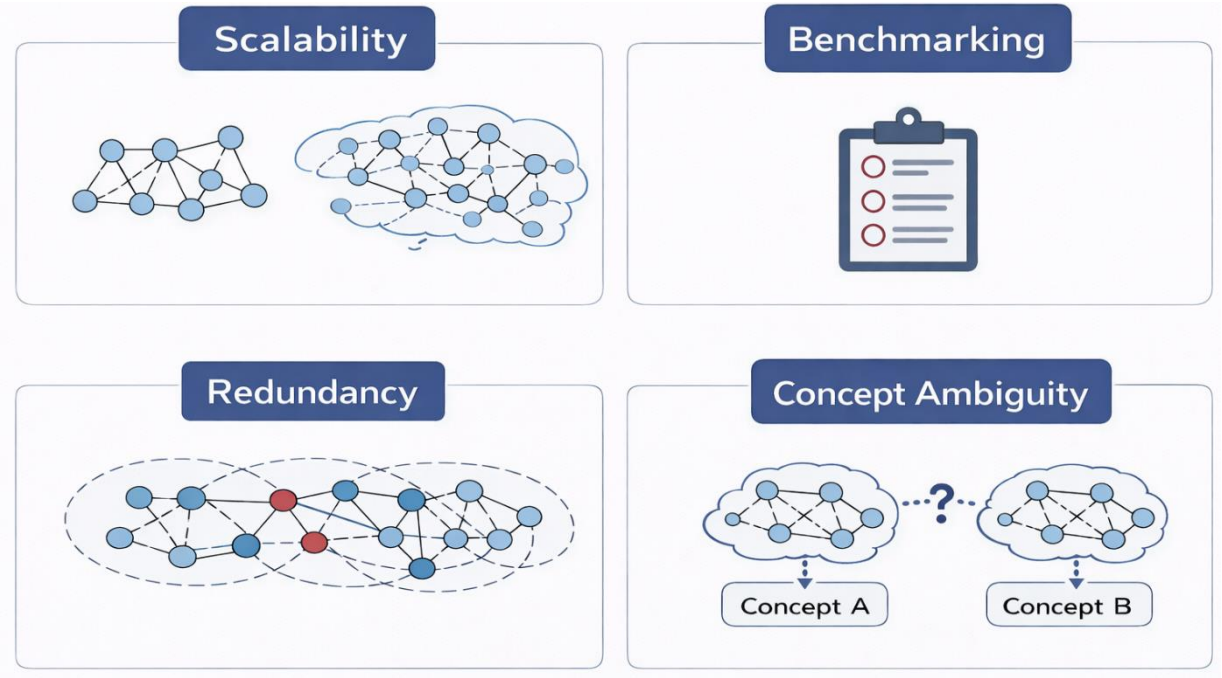


Pattern mining and concept hierarchies are powerful complements to GNN explainability, because:

- **Raise abstraction level:** Node-level saliency or edge importance is low-level and noisy. Mining frequent structural patterns and grouping them into concepts gives explanations at human-meaningful levels.
- **Stability:** Patterns aggregated over many examples are less sensitive to small perturbations.
- **Compression / interpretability:** A compact set of concepts/patterns summarizes many instance-level explanations, reducing cognitive load for humans.
- **Causal / semantic insight.** When patterns align with domain concepts (e.g., protein binding sites, fraud rings), explanations gain actionable semantics beyond raw importance scores.
- **Better evaluation & transfer:** Concepts allow measuring whether the model relies on the *right* recurring motifs.

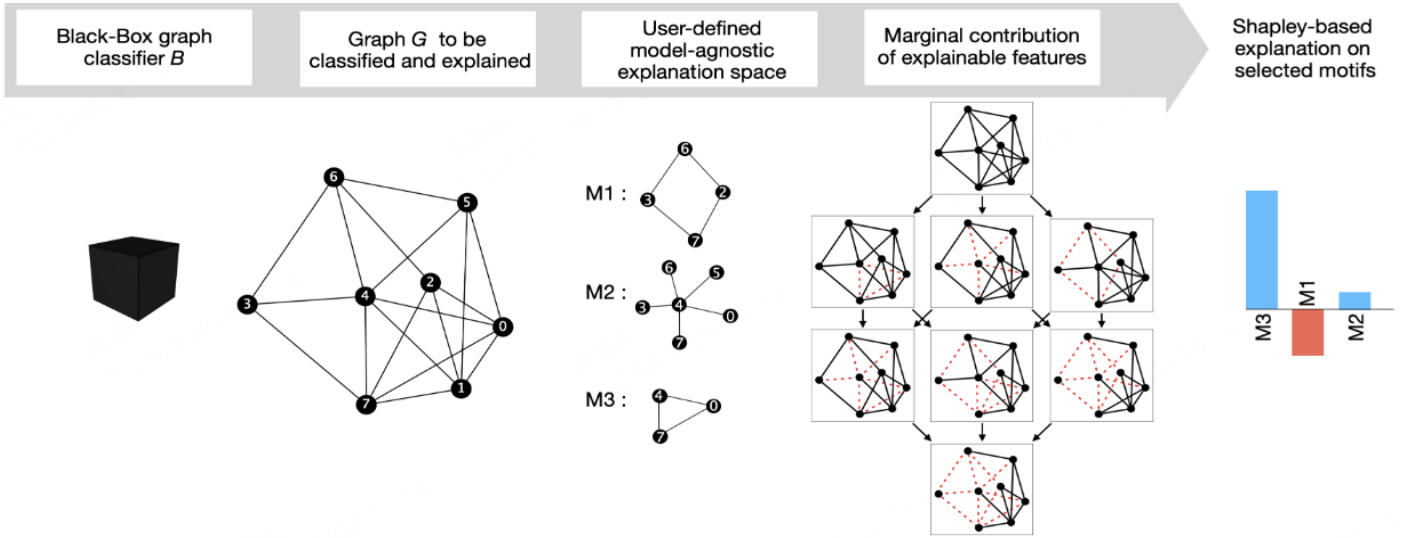
Challenges

- **Scalability of pattern mining:** Exhaustive mining (gSpan, FSG, etc.) explodes with graph size and pattern size. Sufficiency/necessity checks require graph editing and re-evaluating the model many times.
- **Pattern redundancy:** Many overlapping, slightly different patterns that mean the same thing.
- **Benchmarks:** Few standard, widely-accepted metrics for concept-level explanations; how to measure if a concept is “good” (coherent, distinct, useful)?
- **Ambiguity:** A subgraph might belong to multiple concepts or have context-dependent meaning.



Pattern Mining: GRAPHSHAP [IJCNN 23]

Pattern (Motifs): They propose a pattern-based explanation language where a pattern is a motif—a small recurring subgraph structure—rather than explaining predictions with scattered individual edges.

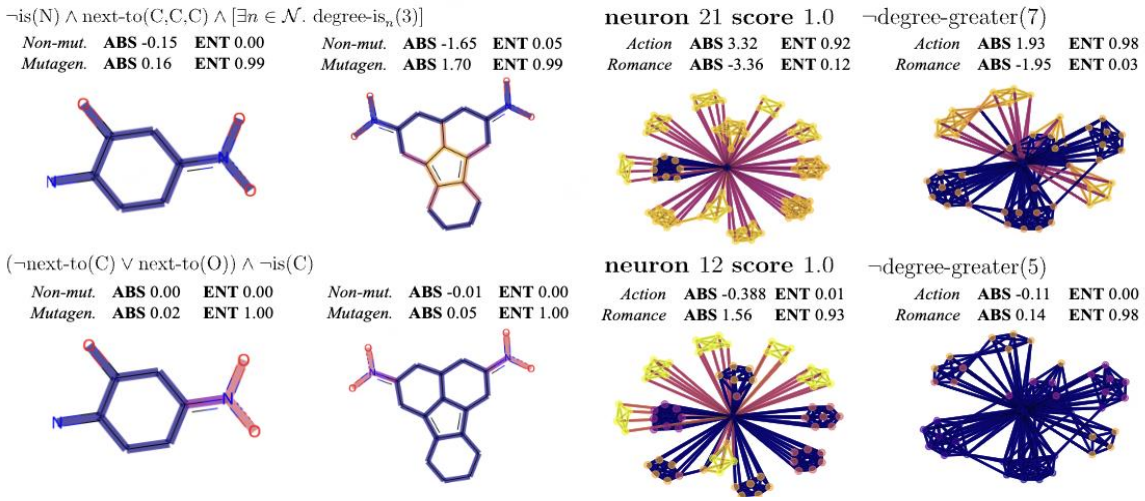


GRAPHSHAP (Shapley Scoring): It decouples the model's feature space (edges) from a human-readable motif space by masking/toggling motifs, querying the black-box classifier, and computing a Shapley-style contribution score for each motif to identify which structural patterns drive the decision.

$$\xi(G, B, M_i) = \sum_{S \subseteq \mathcal{M} \setminus \{M_i\}} \frac{\binom{|\mathcal{M}|+1}{|S|}^{-1}}{|\mathcal{M}|+1} (B(G_S) - B(G_{S \cup \{M_i\}})).$$

Global Concept-Based Interpretability [AAAI 23]

Concept Detectors: These interpretable neurons act as concept detectors, allowing the framework to explain what the model has learned at a global level, rather than just providing a local explanation for a single instance.



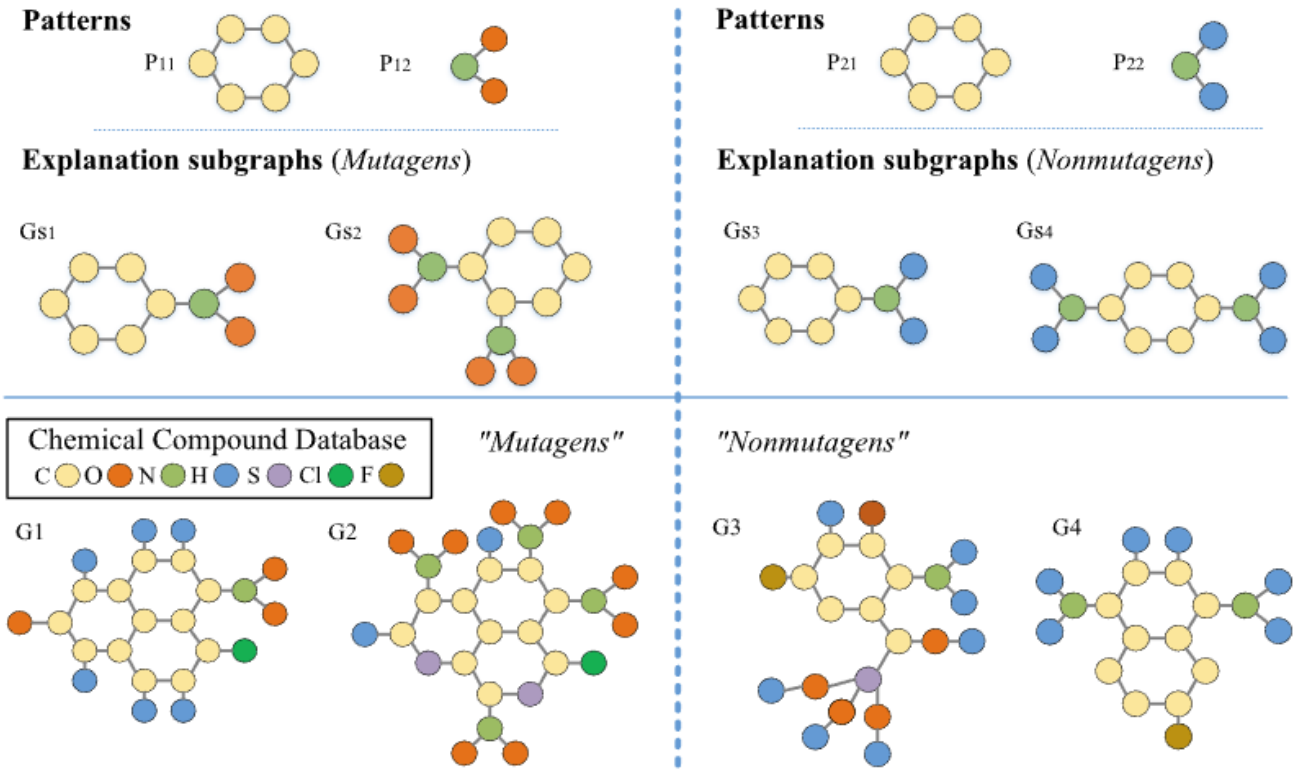
Quantifying Interpretability: The framework measures how well a neuron's activation aligns with a target concept using an overlap-based score. The Div(·) objective includes an IoU-style intersection-over-union term (after thresholding activations by τ) to quantify neuron–concept consistency.

$$\mathrm{Div}\,(\mathbf{a},\mathbf{b},\tau) = -\frac{\sum_i (\mathbf{a} \cap \tau(\mathbf{b}))_i}{\sum_i (\mathbf{a} \cup \tau(\mathbf{b}))_i} \cdot \frac{\mathbf{a} \cdot \tau(\mathbf{b})}{\sum_i b_i},$$

Our Work

Tingyang Chen, Dazhuo Qiu, Yinghui Wu, Arijit Khan, Xiangyu Ke, Yunjun Gao: View-based Explanations for Graph Neural Networks. Proc. ACM Manag. Data 2(1): 40:1-40:27 (2024)

- **GVEX [SIGMOD 24]:** A two-tier explanation framework
- **Motivation:**
  - (1) Large explanation subgraphs hinder explanibaility.
  - (2) Lack of meaningful explanations for domain experts, with limited queryability hindering access and inspection.

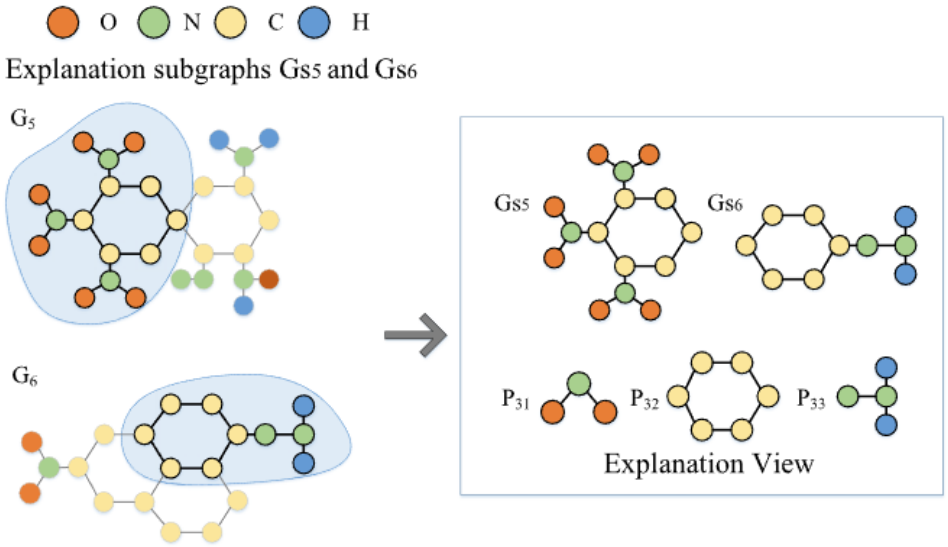


✔ Pattern Matters!

GNN-based drug classification, with graph patterns and induced subgraphs that help understand the results: *"which toxicophore occurs in mutagens?"*

Our Work

- **GVEX [SIGMOD 24]:** A two-tier explanation view framework[1]
  - **How to determine quality of explanation views**
    - Explainability: feature influence ⊕ neighbor diversity
    - Coverage: Numbers of nodes
  - **Efficient algorithm of Explanation View Generation Problem**
    - Greedy, Approximation Algorithm
    - Streaming Extension and Paralllelization



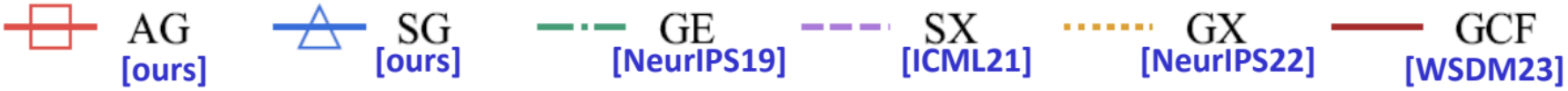
Lower-tier subgraphs: provide factual and counterfactual evidence for predictions

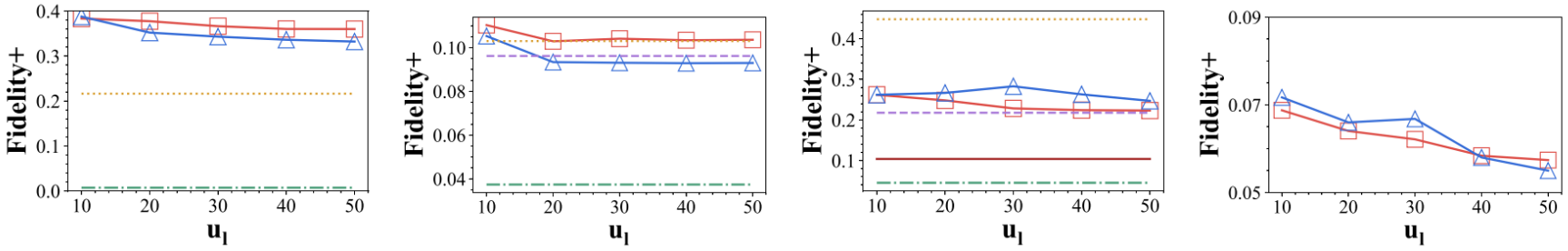
Higher-tier patterns: abstract these subgraphs into common motifs to support efficient search, exploration, and domain alignment.

[SIGMOD 24] Tingyang Chen, Dazhuo Qiu, Yinghui Wu, Arijit Khan, Xiangyu Ke, and Yunjun Gao. 2024. ***View-based Explanations for Graph Neural Networks.***

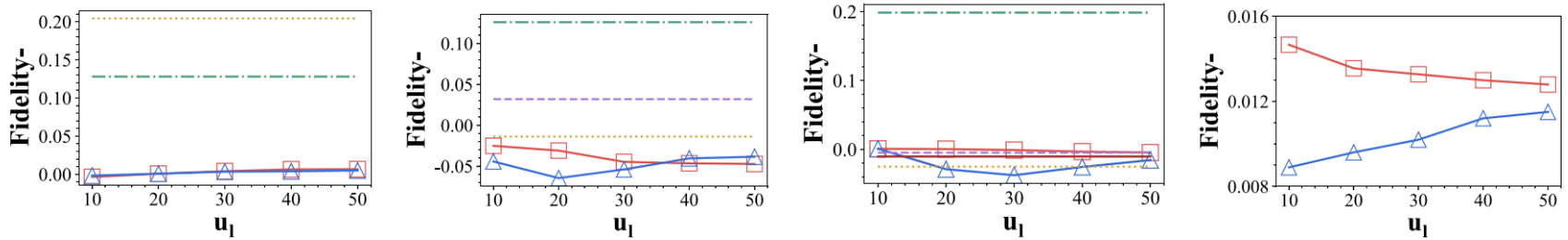
Our Work

- **Experiments:**
  - Fidelity+(Higher is better), Fidelity-(Lower is better)





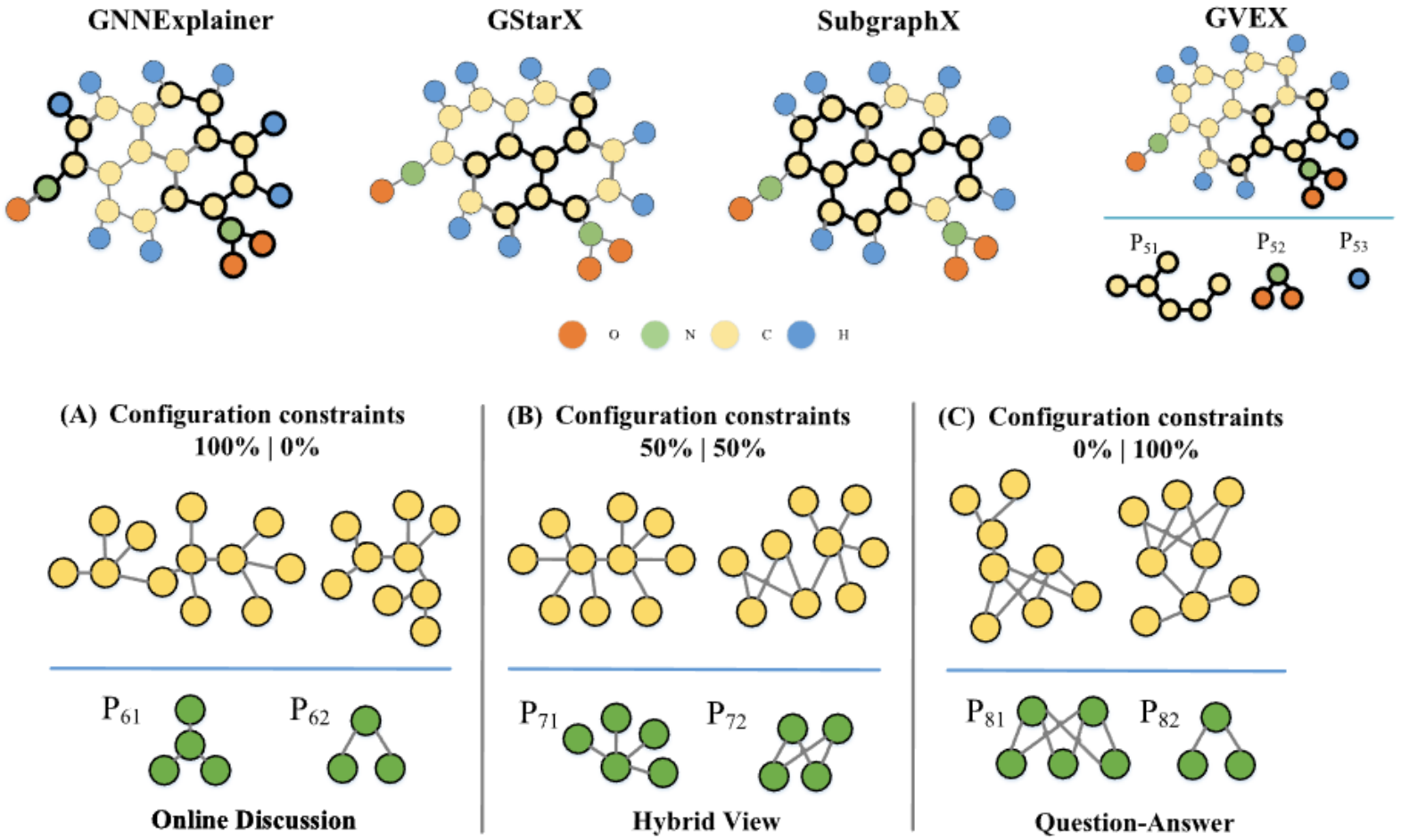
(a) RED (b) ENZ (c) MUT (d) MAL



(a) RED (b) ENZ (c) MUT (d) MAL

Our Work

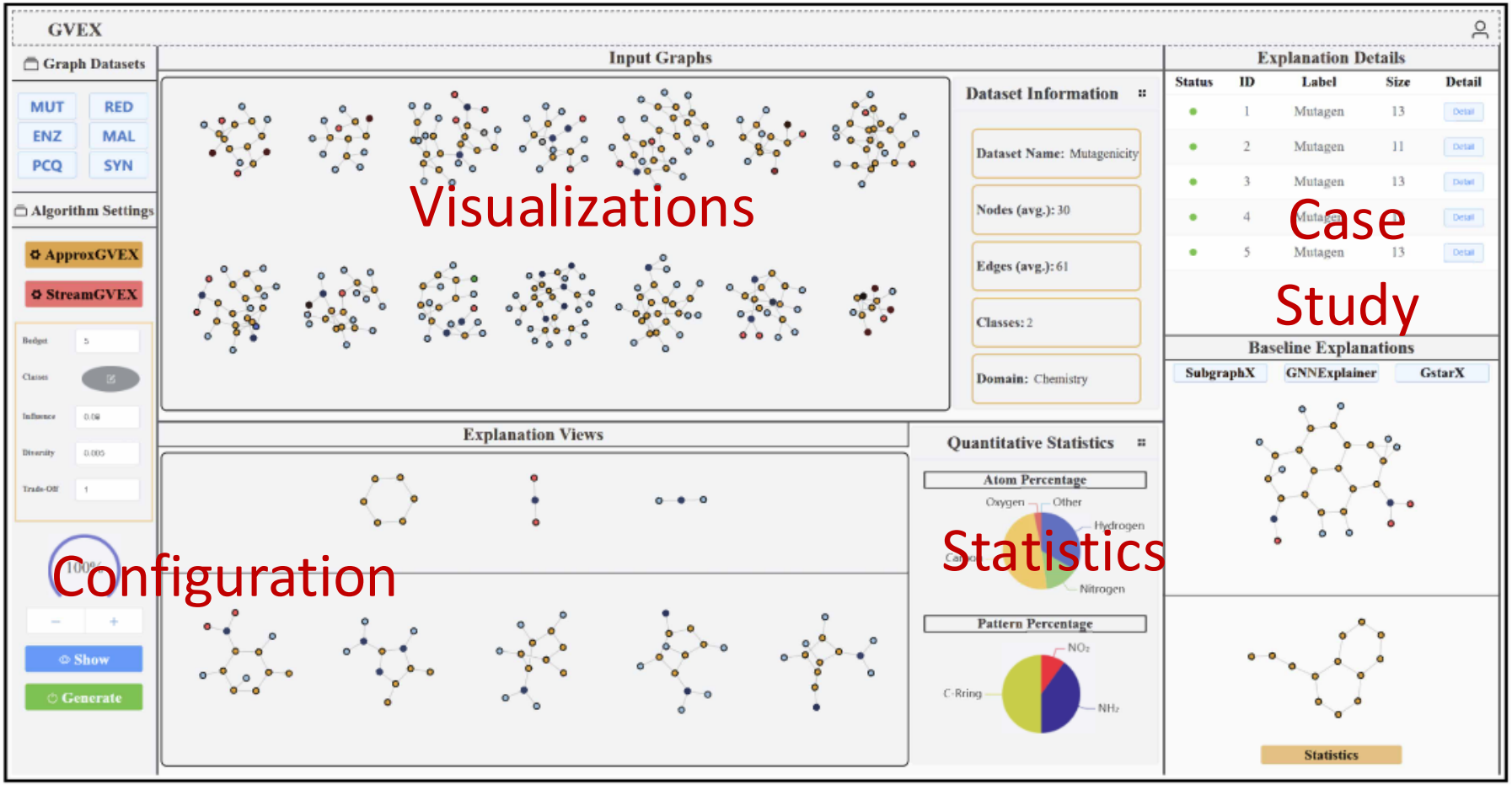
- **Application (drug discovery, social networsk, etc.)**



GNN-based social analysis: REDDITBINARY social network dataset, two classes - online-discussion threads and question-answer threads. Three different configuration scenarios.

Interactiveness

- **GVEX-demo [SIGMOD'24]**[1].



[SIGMOD 24] Tingyang Chen, Dazhuo Qiu, Yinghui Wu, Arijit Khan, Xiangyu Ke, and Yunjun Gao. 2024. ***User-friendly, Interactive, and Configurable Explanations for Graph Neural Networks with Graph Views.***

Future Direction

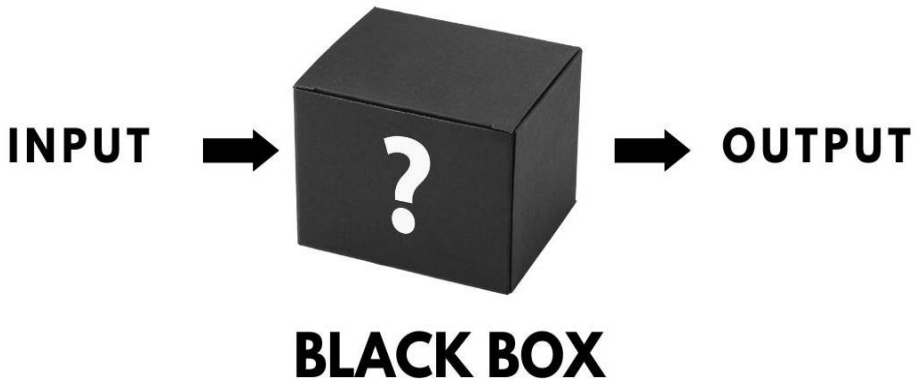
- **In-training concept supervision:** forces model internals to encode human-meaningful concepts so explanations align with what the model uses.
- **Transferable and domain-aligned concepts:** concepts useful in one domain may fail in another; transferability is critical for reuse.
- **Human-centered explanations:** build interactive viewers (hierarchy + prototypes + drill-down) and run small expert studies measuring usefulness for debugging or decision making
- **Bridging to language**: domain experts prefer concise textual rationales tied to patterns

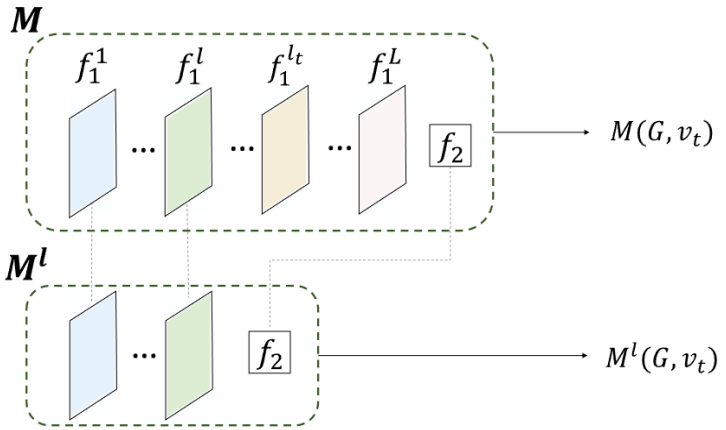
Model-slicing Explanations

Xiangyu Ke

What is model-slicing?

- **From Monolithic to Layer-wise Perspective:**
  - **Traditional XAI:** Treats GNNs as a monolithic function $M(G) \rightarrow y$, focusing only on the input-output relationship.
  - **Model-slicing:** Decomposes GNNs into sequential layer blocks $\{M^1, M^2, \ldots, M^L\}$ to trace the evolution of intermediate representations.



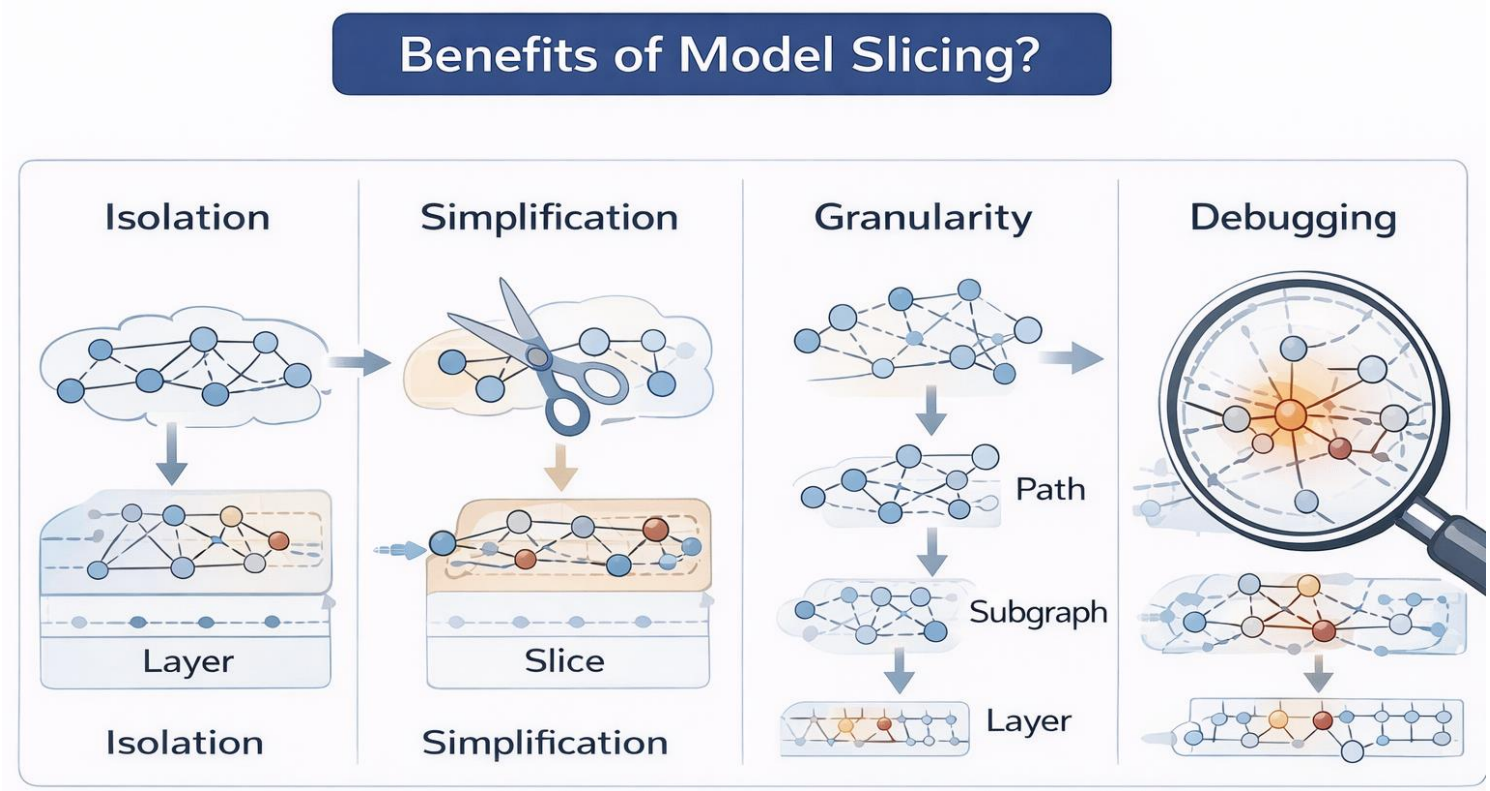


Opaque Process: Focuses solely on the final prediction.
Hidden Dynamics: Internal layer transformations remain invisible.

Decomposed Logic: Sequential layer blocks.
Consistent Evaluation: Shared classifier f_2 for semantic alignment**.**

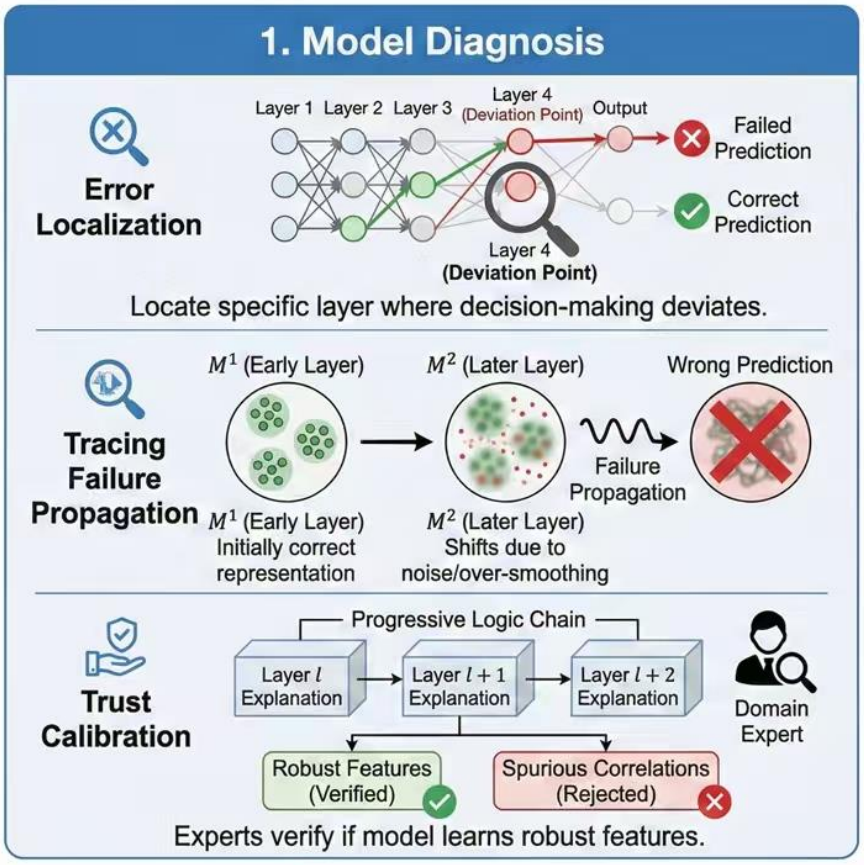
What model slicing buys you?

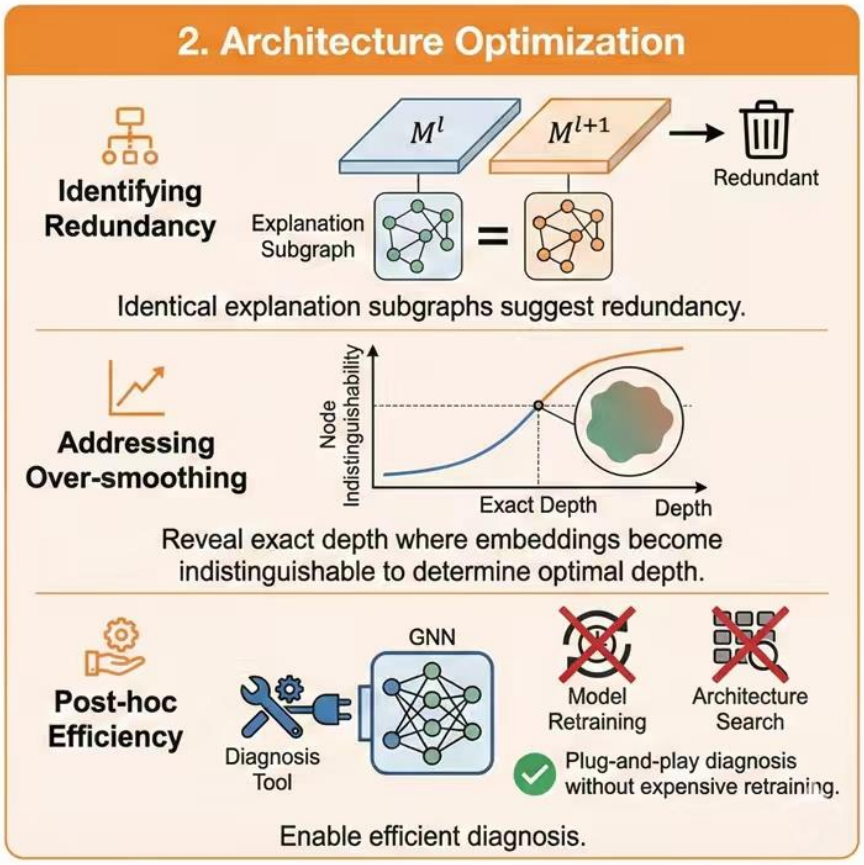
- Isolates *model-internal* causes (message paths, neurons, layers) rather than only input motifs**.**
- Yields compact, causally-testable explanations (ablate slice then observe change).
- Supports multi-granular explanations (path, subgraph, neuron/channel, layer)
- Helps debugging and model repair (pinpoint where bad behavior arises)



What model slicing buys you?

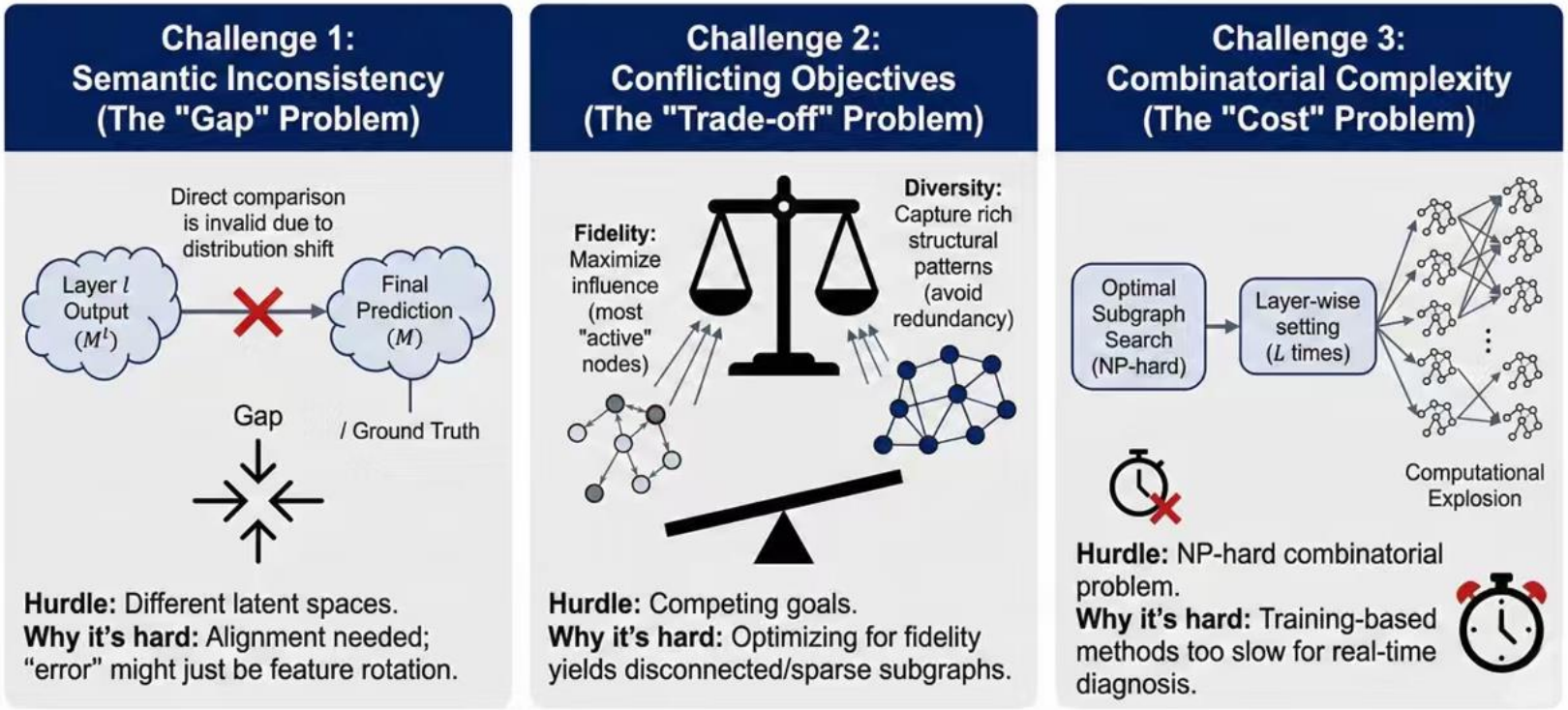
- **Key Applications:**
  - **Model Diagnosis:** Pinpointing "When" and "Where“ by tracing layer-wise decision deviations to localize failure origins.
  - **Architecture Optimization**: Guidance for better design by quantifying structural redundancy to determine optimal model depth.





Why it is hard?

- **Challenges:**
  - **Semantic Inconsistency:** Direct layer-wise comparison is **invalid** due to **latent space misalignment** and distribution shifts.
  - **Conflicting Objectives:** Balancing **prediction fidelity** and **structural diversity** is difficult, often yielding **disconnected** or non-semantic subgraphs.
  - **Combinatorial Complexity:** Layer-wise subgraph search is **NP-hard**, rendering iterative training-based methods **computationally prohibitive** for deep models.

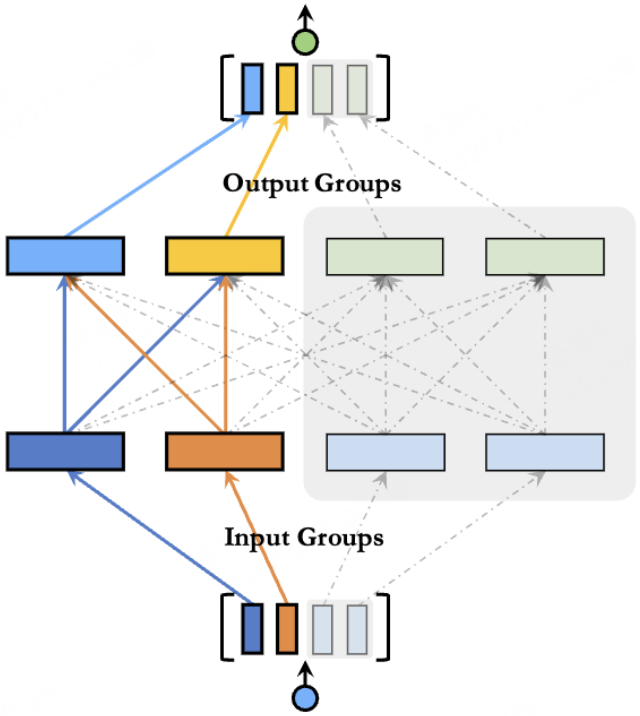


Existing Model Slicing Attempt [VLDB 19]

---- Model Slicing for Supporting Complex Analytics with Elastic Inference Cost and Resource Constraints

Motivation: Deep models used for complex analytics are often too heavy to run under fluctuating workloads and strict resource budgets, so we need a way to dynamically trade off accuracy and inference cost within a single model.

Model Slicing trains a network so that any prefix slice of each layer's channels forms a valid sub-model. Concretely, channels in each layer are ordered by importance and shared across slices: a small slice uses only the first few channels, while a larger slice reuses them and adds more channels on top.
During training, the model is jointly optimized under multiple slice widths, so that at inference time the system can select an appropriate slice size on the fly according to current latency, FLOPs, or memory constraints, without retraining or switching models.



Left figure illustrates **the core idea** of model slicing: each layer is organized so that its channels can be sliced from left to right, yielding multiple nested sub-models.

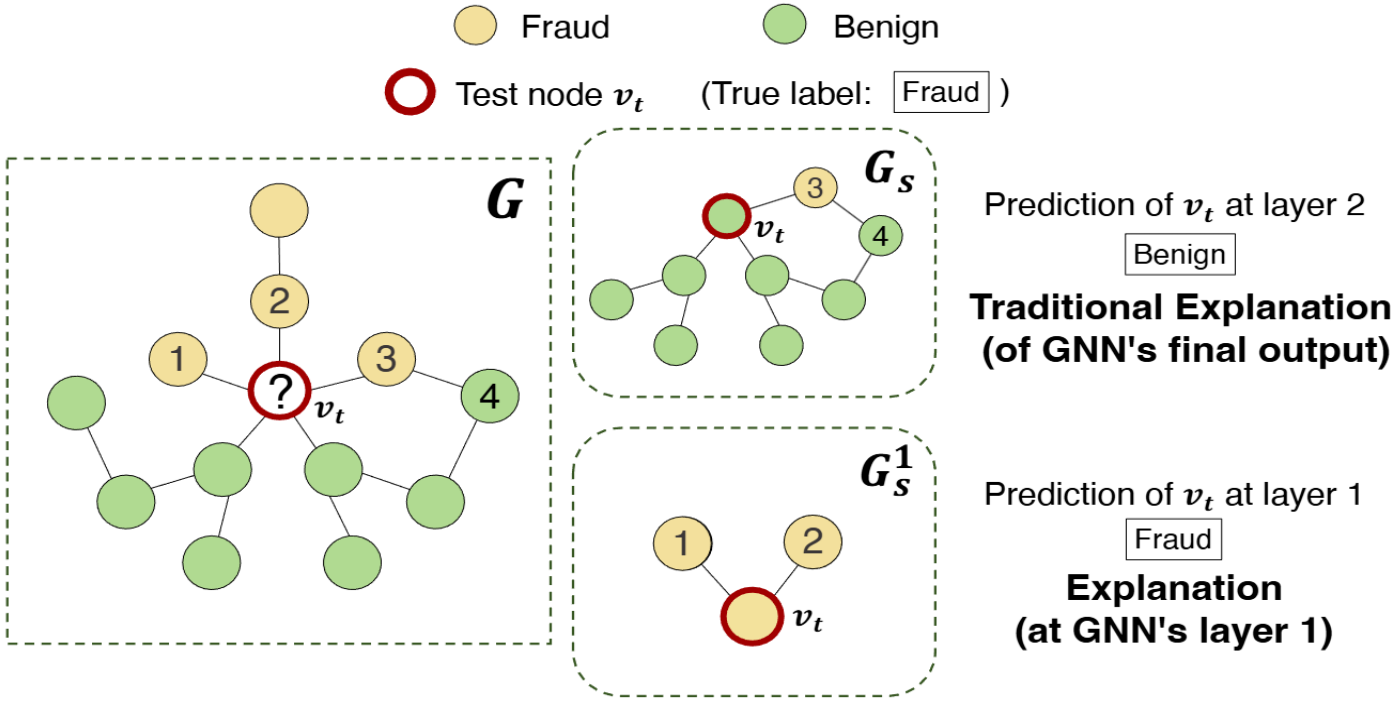
Smaller slices (fewer channels) incur lower inference cost but reuse the same parameters as larger slices, which simply append additional channels for higher accuracy.

At runtime, the system chooses an appropriate slice according to the current resource budget.

Our Work: SliceGX

Cibo Yu, Tingting Zhu, Tingyang Chen, Yinghui Wu, Arijit Khan, and Xiangyu Ke, "SliceGX: Layer-wise GNN Explanation with Model-slicing", in Proc. of The Web Conference 2026.

- **SliceGX [WWW 26]:** A Layer-wise Progression
- **Motivation:**
  - (1) **Intermediate Opacity**: Existing methods overlook how explanations evolve across layers.
  - (2) **Diagnostic Gap**: Unable to pinpoint which specific layer causes model failure.

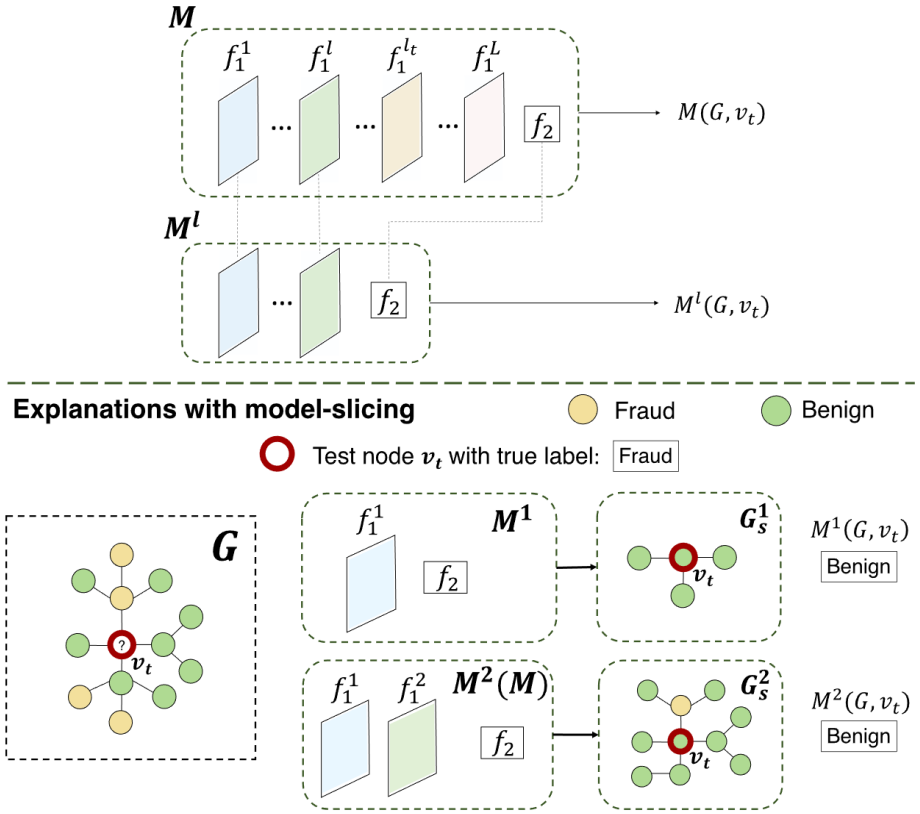


Generating layer-wise explanations for GNN diagnosing for *spam review*

Our Work: SliceGX

- **SliceGX [WWW 26]:** A Layer-wise Progression

Goal: To provide **progressive**, **layer-wise** explanations for model diagnosis and architecture optimization.



Model-Slicing: Decomposes GNNs into layer blocks to trace intermediate representation transformations.

Bi-criteria Metric: Optimizes both message-passing influence and embedding diversity to ensure high-quality, unbiased subgraphs.

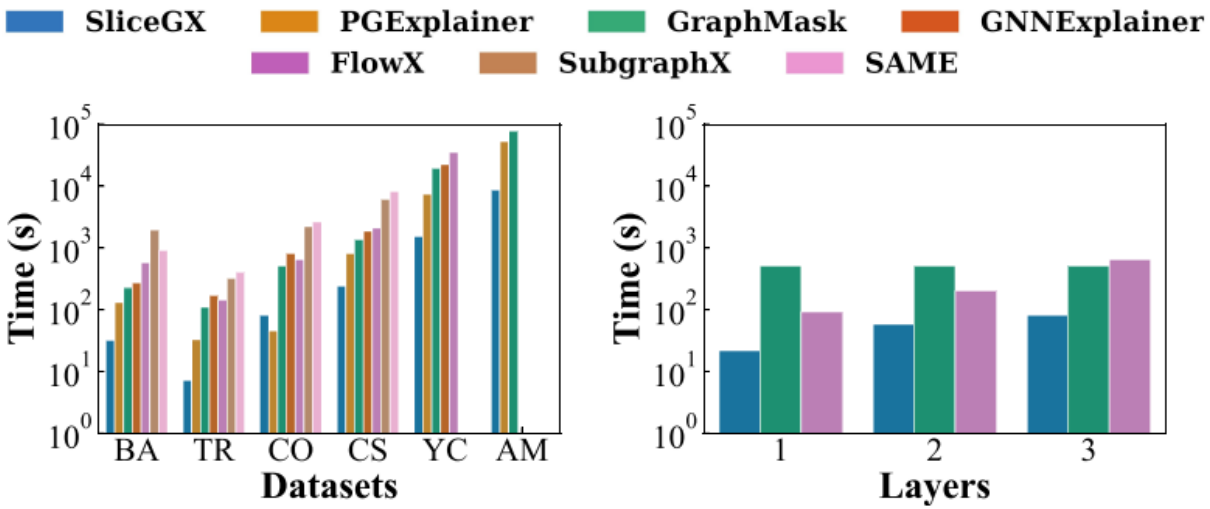
Post-hoc Algorithms: Training-free generation with provable $\frac{1}{2}$ −approximation guarantees for efficient discovery.

[2] Cibo Yu, Tingting Zhu, Tingyang Chen, Yinghui Wu, Arijit Khan, and Xiangyu Ke. WWW '2026. **SliceGX: Layer-wise GNN Explanation with Model-slicing.**

Our Work

- **Experiments:**
  - Fidelity+(Higher is better), Fidelity-(Lower is better)

	BA-shapes		Tree-Cycles		Cora		Coauthor CS		YelpChi		Amazon	
Explainers	Fid+ (↑)	Fid- (↓)	Fid+ (↑)	Fid- (↓)	Fid+ (↑)	Fid- (↓)	Fid+ (↑)	Fid- (↓)	Fid+ (↑)	Fid- (↓)	Fid+ (↑)	Fid- (↓)
GNNExplainer	0.2665	0.7264	0.6065	0.6611	0.1088	0.8173	0.4302	0.1525	0.2041	0.4238	—	—
PGExplainer	0.2763	0.6697	0.3689	0.2551	0.2739	0.2041	0.3217	0.2815	0.3482	0.4255	0.0421	0.8469
GraphMask	0.4302	0.3580	0.1243	0.0280	0.1260	0.7202	0.2341	0.4512	0.1245	0.7241	0.0012	0.7187
SubgraphX	0.2307	0.4018	0.7012	0.0782	0.4748	0.1003	0.5142	0.1451	—	—	—	—
SAME	0.0303	0.6662	0.5124	0.4920	0.3984	0.2522	0.3516	0.1242	—	—	—	—
FlowX	0.4682	0.7062	0.5052	0.2583	0.6430	0.7471	0.5189	0.1432	0.3014	0.2414	—	—
SliceGX	**0.6918**	**0.0670**	**0.8047**	**0**	**0.7117**	**0.0531**	**0.7003**	**0.0341**	**0.5014**	**0.1294**	**0.1724**	**0.4521**



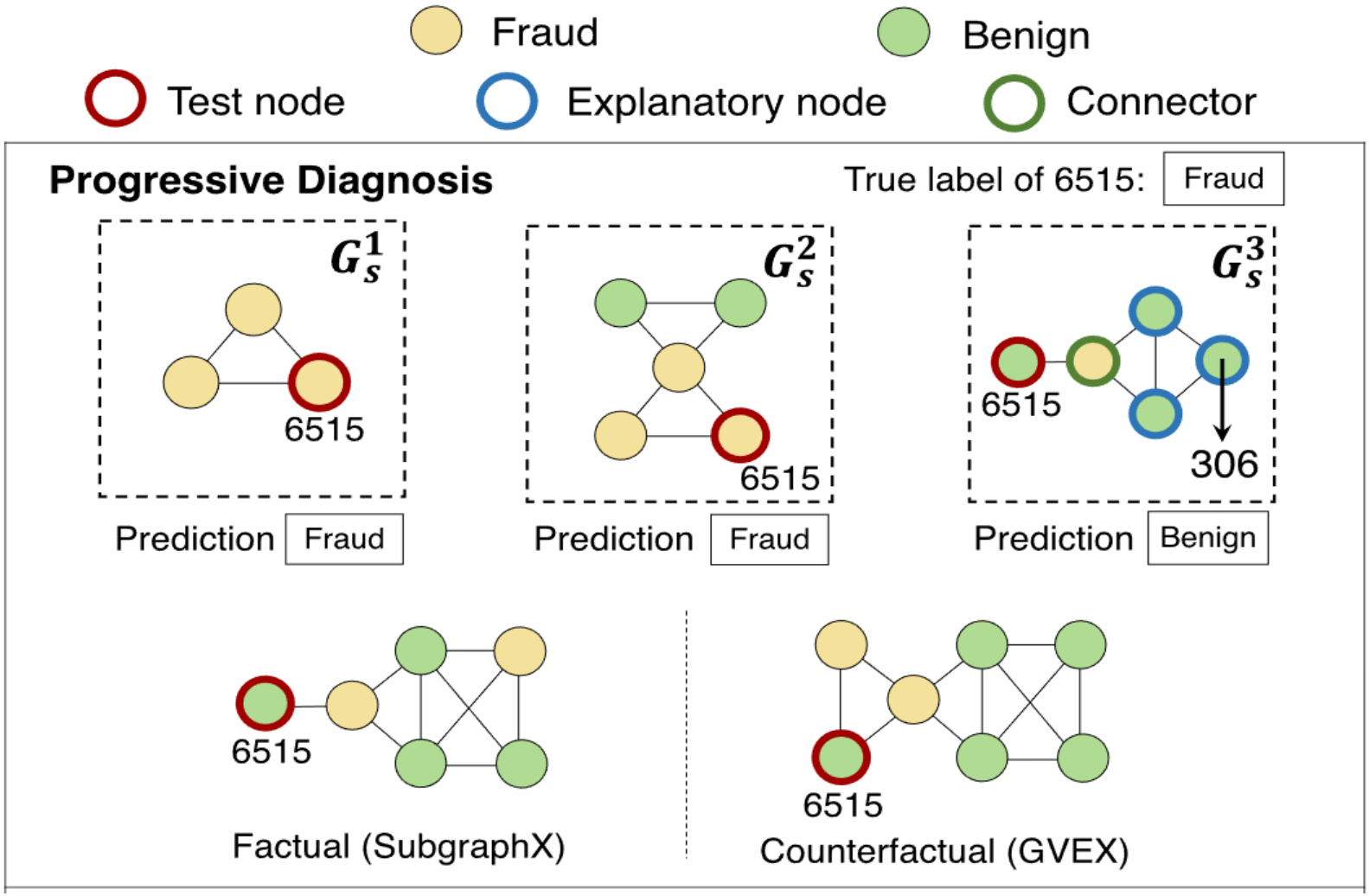
(a) Efficiency: Overall

(b) Efficiency: Varying source layer

Efficiency

Our Work: SliceGX

- **Case study (progressive diagnose)**



SliceGX reveals how a prediction is formed **layer-by-layer**, whereas SubgraphX and Gvex only show what the final result is.

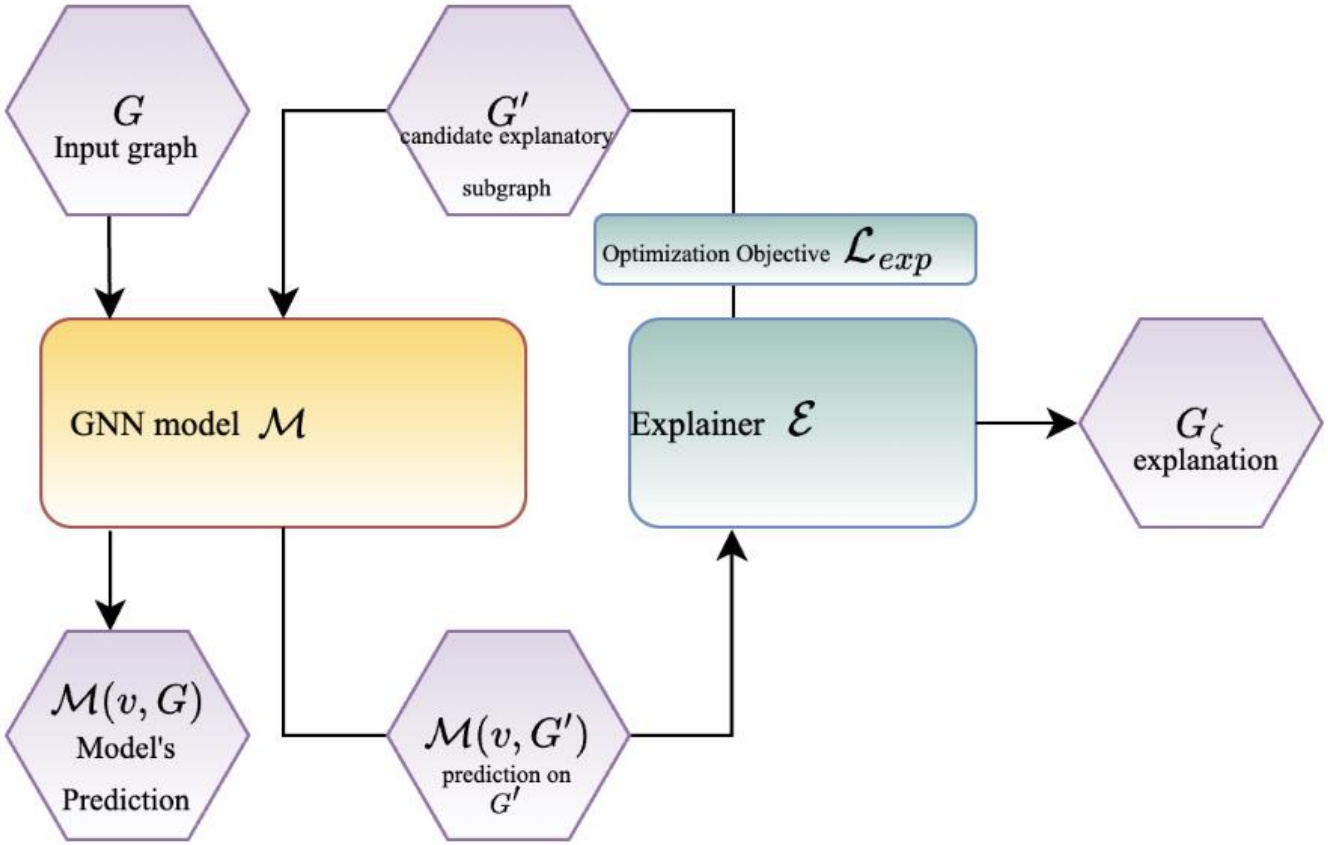
Future Directions in Layer-wise XGNN

- **Transferability & cross-architecture slices:** Slices useful across models (e.g., move from GAT to GCN) or datasets increases reusability.
- **Slice-driven compression and pruning** : Slices reveal which paths/neurons are rarely used—useful for compression without harming task performance.
- **Fairness & privacy**: Slices might expose sensitive patterns or amplify biases
- **Coalition-aware slices (for synergistic effects):** Many decisions arise from joint contributions that single-element slices miss
- **Theoretical foundations: minimality, uniqueness, and approximation bounds:** practitioners need guarantees: is a slice minimal/unique and how good are approximations?
- **Human-centered evaluation:** explainability is useful only if humans understand and can act on slices.

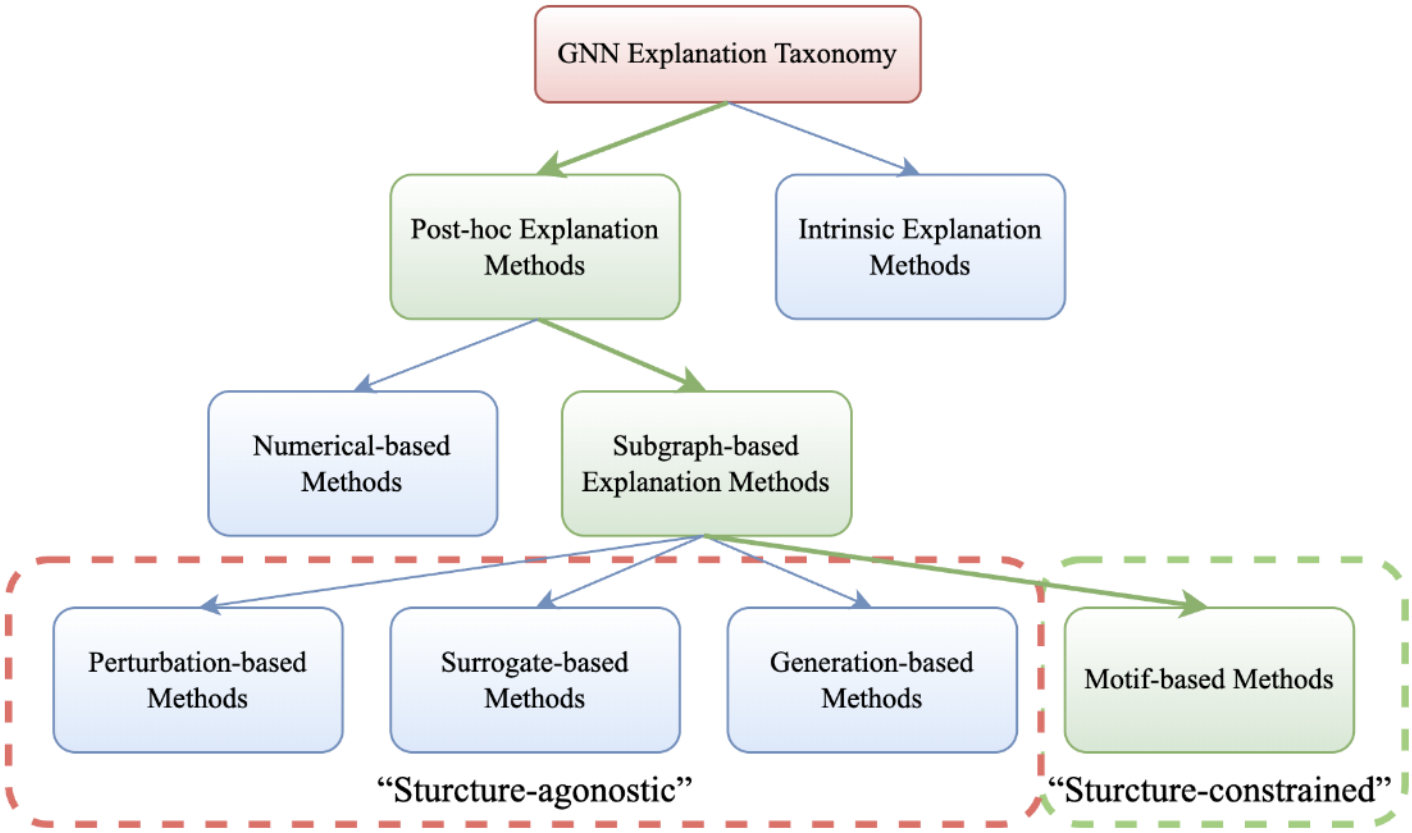
Robust Explanation

Yinghui Wu

GNN Explanation: A general recipe



A taxonomy of SOTA GNN Explainers

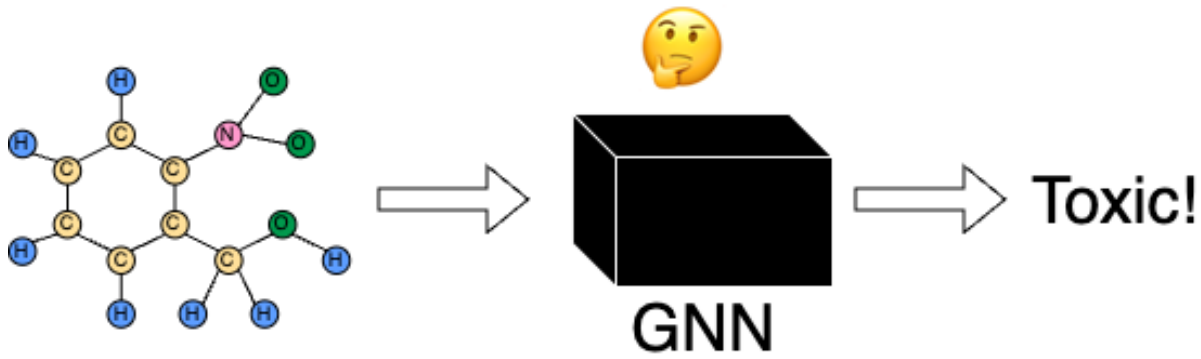


Dazhuo Qiu, Mengying Wang, Arijit Khan, and Yinghui Wu, "Generating Robust Counterfactual Witnesses for Graph Neural Networks", in Proc. of IEEE International Conference on Data Engineering 2024.

Robust GNN Explanation

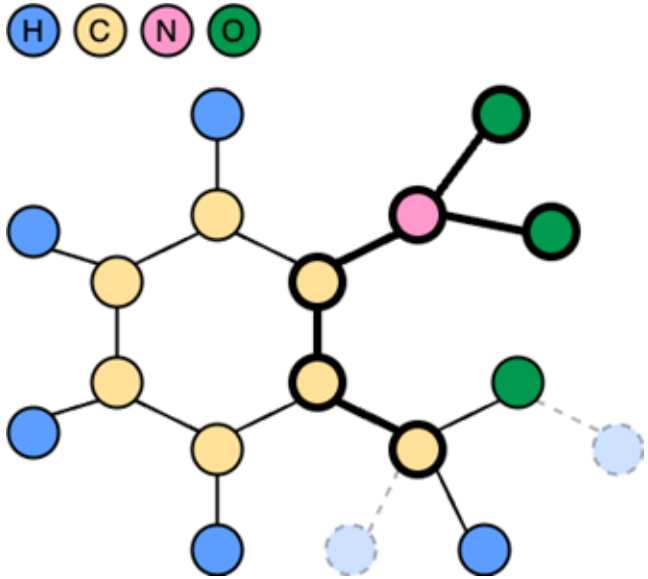
- ***"Black-Box" GNNs***:
  - The inference of GNN models are black-box.
  - Hard to understand which part of the input causes the results.

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



The need for Robust Explanation

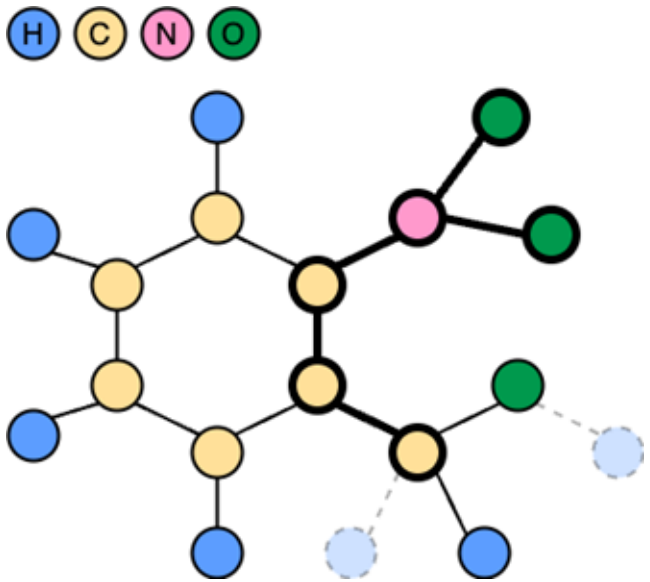
- ***Factual Explanation (Witness)***:
  - M(v, G) = M(v, Gs) = l
- ***Counterfactual Explanation (CW)***:
  - M(v, G) ≠ M(v, G\Gs) ≠ l
- *Robust Explanation*:
  - Gs should remain a *consistent*, *stable* structure under various disturbance.



Multicriteria GNN Explanation

- ***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

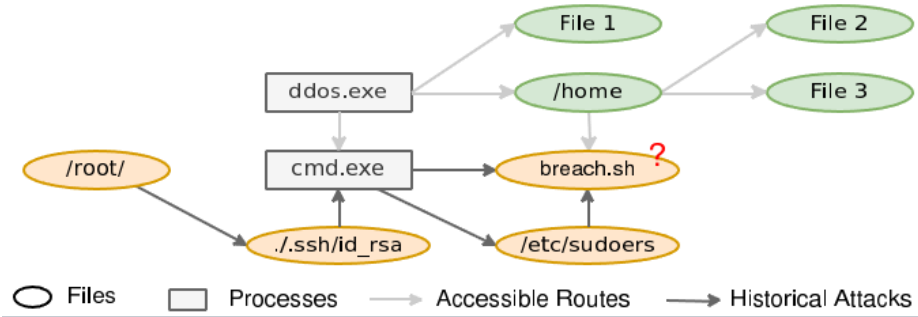
  consistent under k-disturbance.



We should consider all three aspects

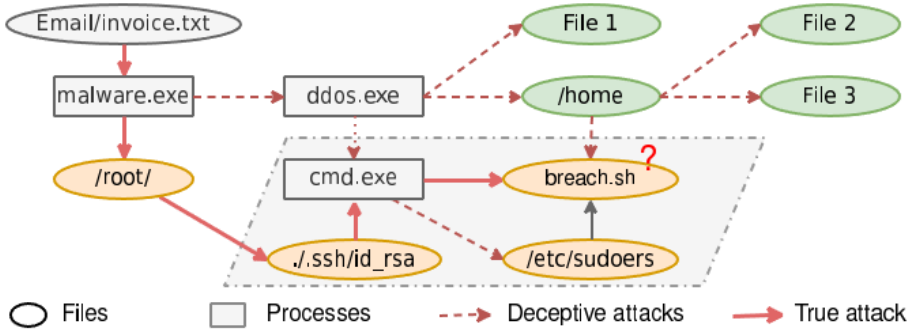
Need for Robust Explanations: Cyberattacks

- Interpreting cyberattacks under agile attacking strategies



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:

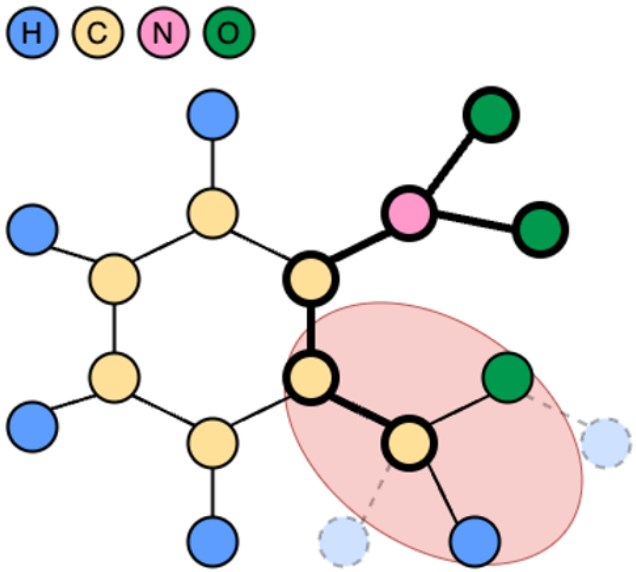
- **Phase 1: Deception Attacks**: Conduct deceptive but harmless attacks to Induce false invulnerable classification on target.
- **Phase 2: True Attack**: attack by exploiting reduced defenses on target.

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

Verification problem

- ***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)



Size of remaining graph

$$O(L|G'|(d_m \log d_m + LF(|E| + |V|F)))$$

# of classes

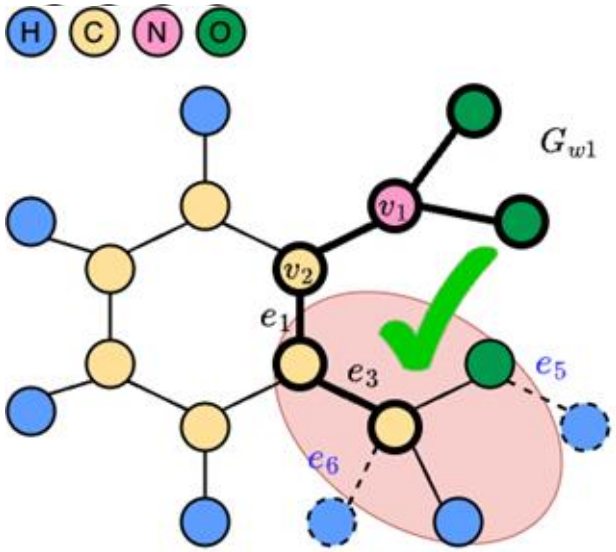
Sorting cost of a single node

One time APPNP inference cost

Generation problem

- ***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.



Experimental Evaluation

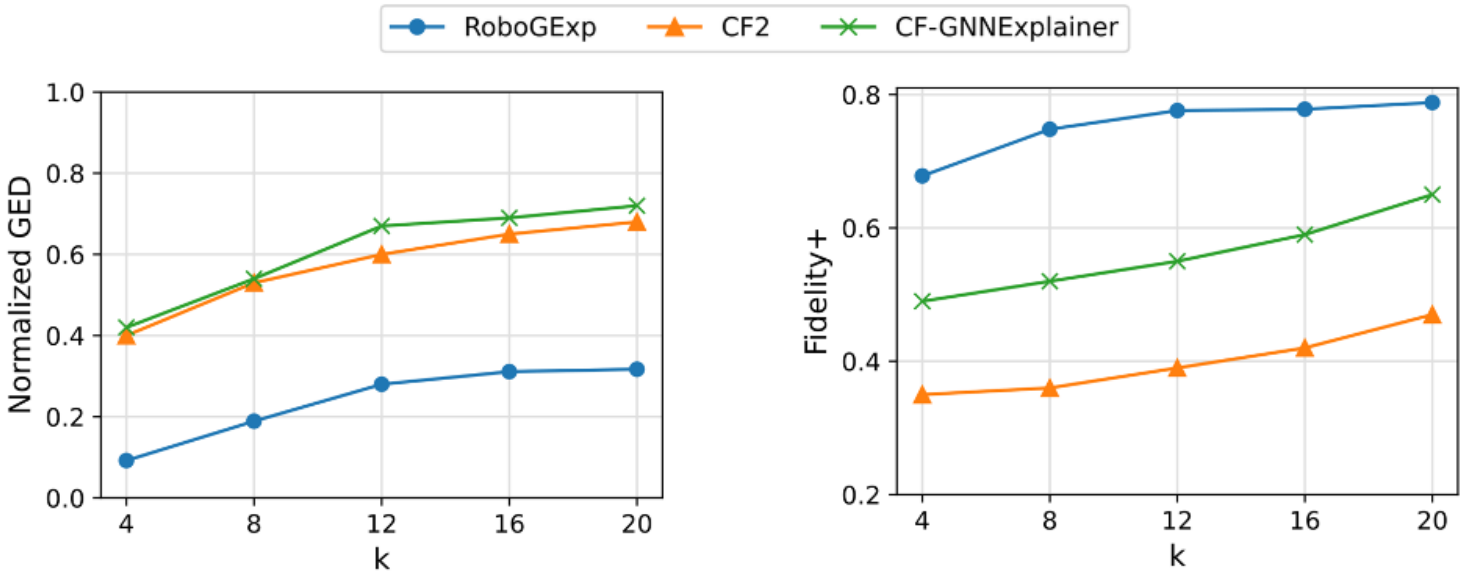
Dataset	# nodes	# edges	# node features	# class labels

	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

RoboGExp	✓	✓	✓

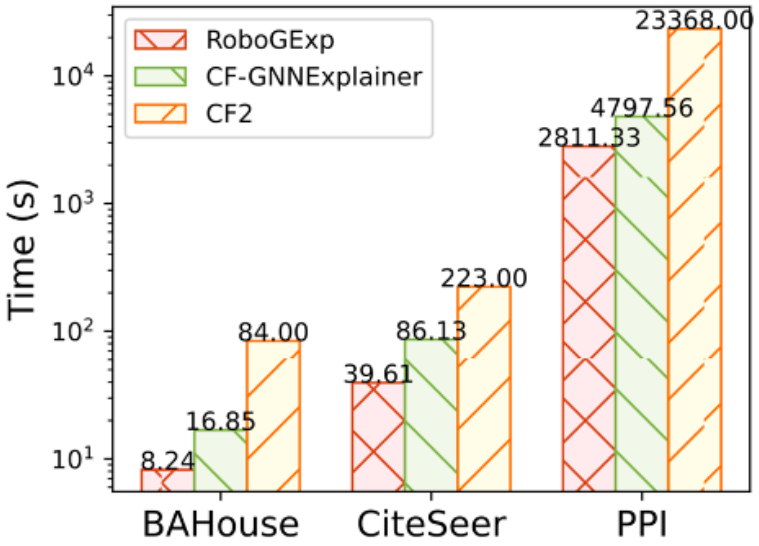
- ***Normalized GED***:
  - Robustness facilitates the consistency of the explanation under disturbance.
- ***Fidelity+ and Fidelity-***:
  - Verification procedure ensures a high fidelity performance.
- ***Size***:
  - RoboGExp integrate the explanation of each test node.

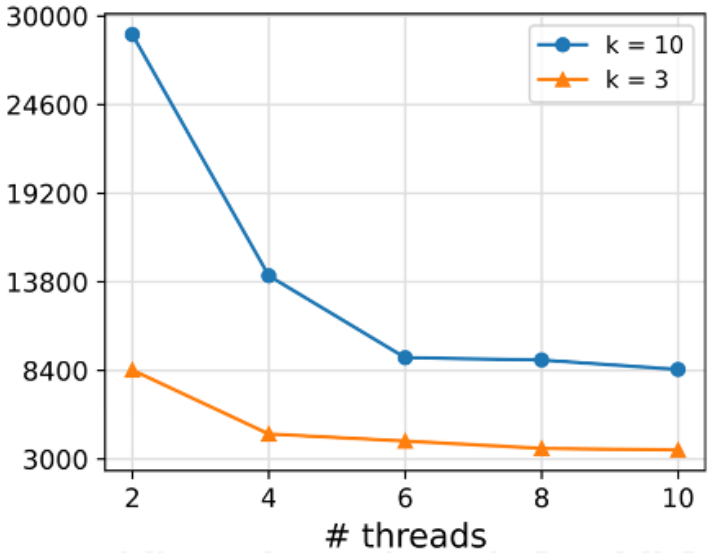
Effectiveness



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

Efficiency





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

End of First Half

Multi-goal Driven Explanation

Yinghui Wu

Interpreting GNNs: Why multi-criteria?

- ML/AI Interpretability is task/user-specific
  - Single-metric explainers may produce one-sided, biased perspective

- Call for multiple high-quality explanations capturing different trade-offs
  - Fidelity → large subgraphs, hard to inspect
  - Conciseness → small subgraphs, may miss important nodes
  - Counterfactual → informative but bloated explanations
  - Linear combinations are sensitive to weights and bias

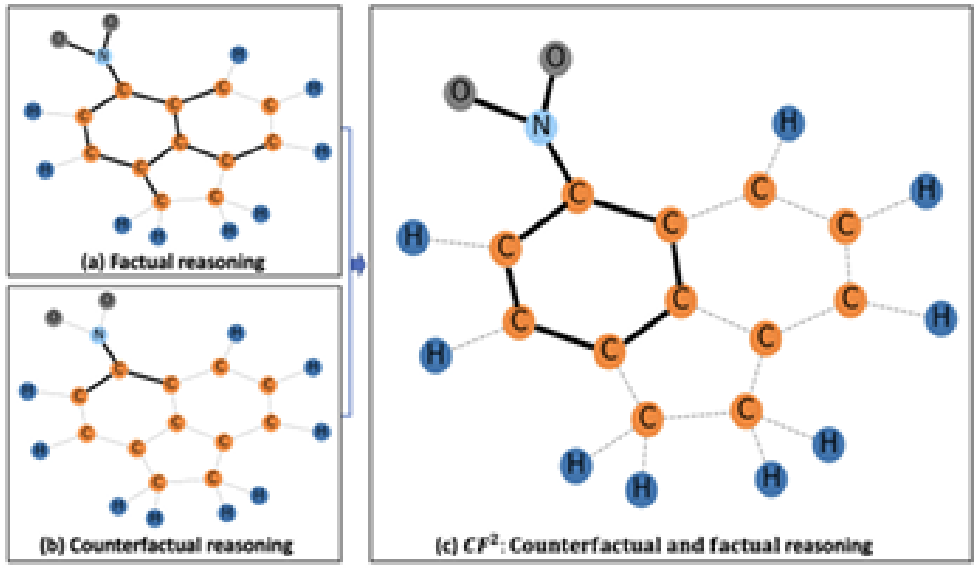
Skyline Explanations: Pareto-Optimality

Symbol	Measure	Equation	Range	Description
fac	factual	$\mathcal{M}(v,G) = \mathcal{M}(v,G_\zeta)?$	{true,false }	a Boolean function
cfac	counterfactual	$\mathcal{M}(v,G) \neq \mathcal{M}(v,G \setminus G_\zeta)?$	{true,false }	a Boolean function
fdl^+	fidelity^+	$Pr(M(v,G)) - Pr(M(v,G \setminus G_\zeta))$	$[-1,1]$	the larger, the better
fdl^-	fidelity^-	$Pr(M(v,G)) - Pr(M(v,G_\zeta))$	$[-1,1]$	the smaller, the better
conc	conciseness	$\frac{1}{N}\sum_{i=1}^{N}(1 - \frac{\lvert E(G_\zeta)\rvert}{\lvert E(G)\rvert})$	$[0,1]$	the smaller, the better
shapley	Shapley value	$\phi(G_\zeta) = \sum_{S \subseteq P \setminus \{G_\zeta\}} \frac{\lvert S\rvert!(\lvert P\rvert - \lvert S\rvert - 1)!}{\lvert P\rvert!} m(S, G_\zeta)$	$[-1,1]$	total contribution of nodes in G_ζ

- Return a set of non-dominated explanations
- Each explanation is best in at least one metric
- Users choose among diverse, high-quality options

CF2 (WWW 2022): Counterfactual & Factual Explanations

- Goal: Learn explanations that are both factual and counterfactual
- Jointly learns edge and feature masks
- Optimizes weighted objective over factual + counterfactual reasoning
- Strength: clear semantics for 'why' and 'what-if-not'



CF2: Intuition & Workflow

- Workflow:
  - Input graph → learn masks → evaluate factual & counterfactual objectives
  - Output: one concise explanation subgraph
  - Limitation: single explanation; sensitive to weight tuning

Condition for Factual Reasoning :

$$\arg\max_{c \in C} P_\Phi(c \mid A_k \odot M_k, X_k \odot F_k) = \hat{y}_k$$

Condition for Counterfactual Reasoning :

$$\arg\max_{c \in C} P_\Phi(c \mid A_k - A_k \odot M_k, X_k - X_k \odot F_k) \neq \hat{y}_k$$

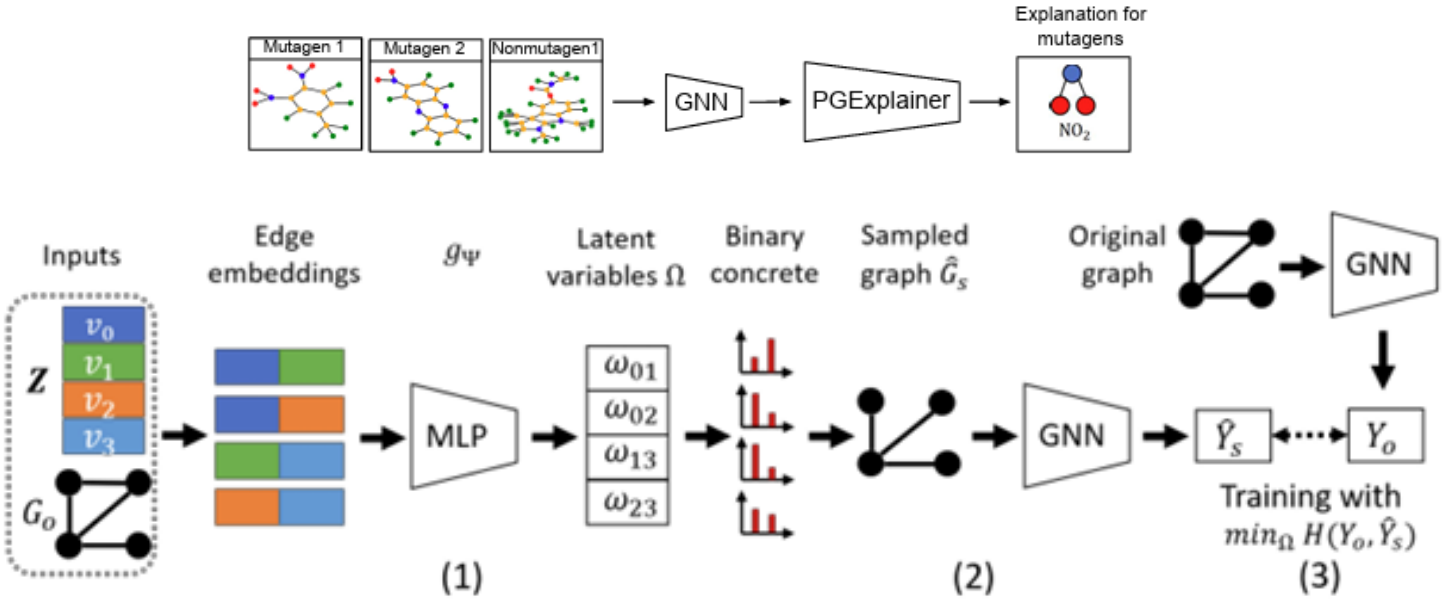
$$\text{minimize } C(M_k, F_k)$$

$$\text{s.t., } S_f(M_k, F_k) > P_\Phi(\hat{y}_{k,s} \mid A_k \odot M_k, X_k \odot F_k),$$

$$S_c(M_k, F_k) > -P_\Phi(\hat{y}_{k,s} \mid A_k - A_k \odot M_k, X_k - X_k \odot F_k)$$

PGExplainer (NeurIPS 2020): Parameterized Explainer

- Goal: Learn a reusable explainer model for GNNs
- MLP predicts edge importance conditioned on node embeddings
- Strength: Trained once, generally applied to many nodes/graphs; fast inference-time explanations



PGExplainer: Intuition & Workflow

- Workflow:
  - Pretrain explainer → generate edge mask → extract subgraph
  - Optimizes mutual information with GNN predictions
  - Limitation: training bias affects quality; learning overhead

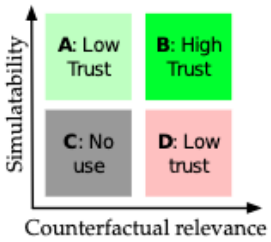
$$\min_{\Omega} \mathbb{E}_{\epsilon\sim\text{Uniform}(0,1)} H(Y_o, \hat{Y}_s) \approx \min_{\Omega} -\frac{1}{K}\sum_{k=1}^{K}\sum_{c=1}^{C} P(Y_o = c)\log P(\hat{Y}_s = c)$$

$$=\min_{\Omega} -\frac{1}{K}\sum_{k=1}^{K}\sum_{c=1}^{C} P_\Phi(Y = c|G = G_o)\log P_\Phi(Y = c|G = \hat{G}_s^{(k)}),$$

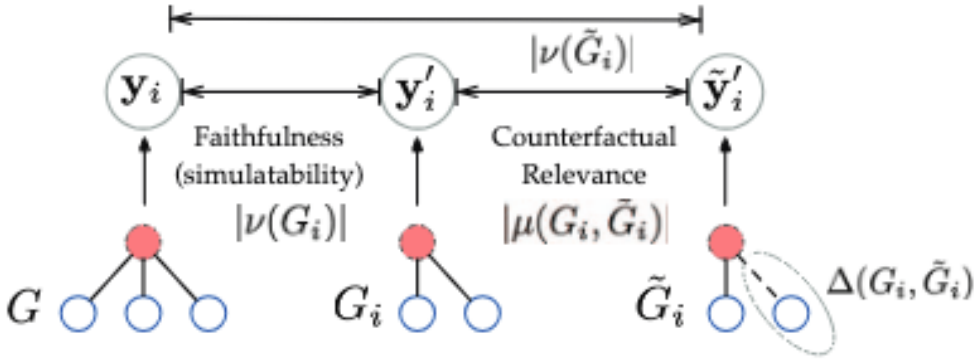
Approximation strategies: Use cross entropy H to approximate conditional entropy; Use Monte Carlo to approximately optimize the object over a set of sampled graphs

MOExp (ICDM 2021): Multi-Objective Explanations

- Goal: Pareto-optimal explanations over two objectives
- Optimizes factual (simulatability) and counterfactual relevance
- Returns a Pareto set instead of one explanation
- Strength: exposes trade-offs explicitly

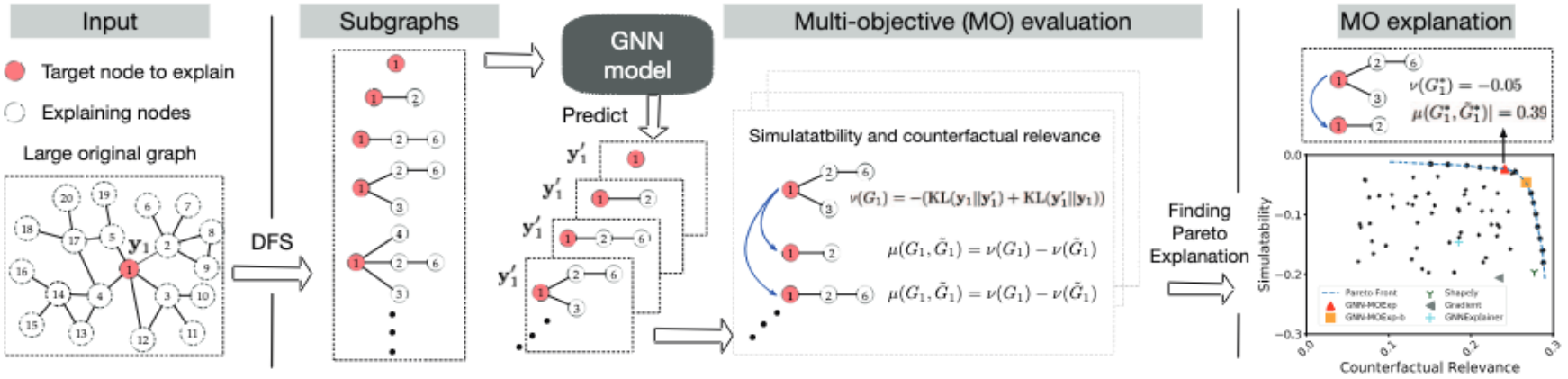


High simulatability or counterfactual relevance is a necessary but not a sufficient condition for an explanation to be perceived as trustworthy by humans.



MOExp: Intuition & Workflow

- Workflow:
  - Search subgraphs → evaluate two objectives → Pareto front
  - User inspects multiple explanations
  - Limitation: number of explanations can be large

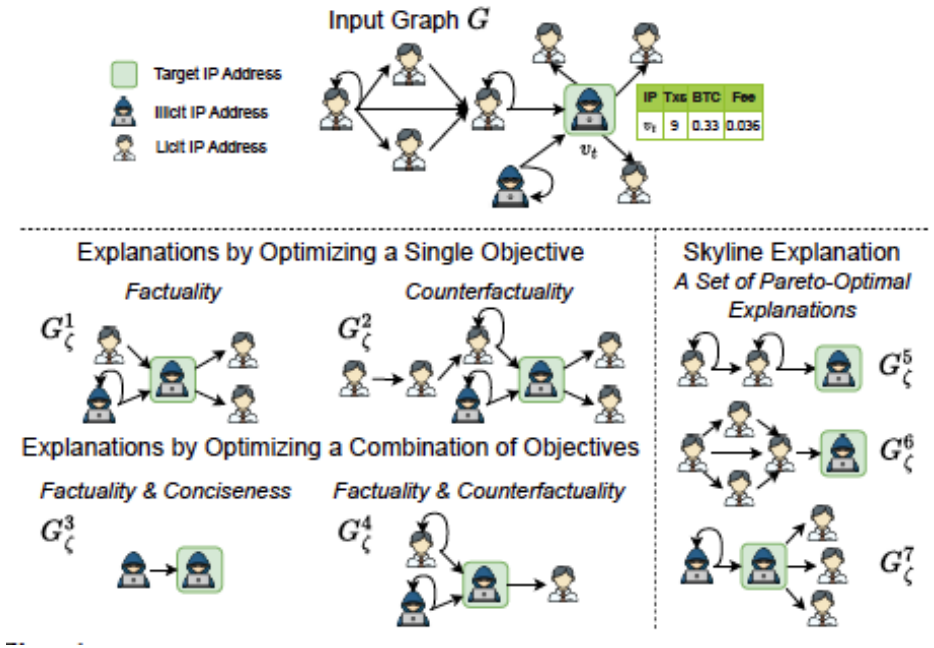


Comparison

- CF2: factual + counterfactual via weighted objective → one explanation
- PGExplainer: reusable parameterized explainer → fast, single, general-purpose objective
- MOExp: Pareto front over two objectives → multiple explanations
- Trade-off: expressiveness vs. scalability vs. inspection cost

SkyGExp (ICDE 2026): Skyline Subgraph Explanations

- Goal: Pareto-optimal explanations over any k objectives
- Learning-free, fast subgraph generation;
  scale to billion-level graphs
- Returns a Pareto set instead of one explanation
- Exposes trade-offs explicitly



Fraud detection, cybersecurity, drug discovery, and recommendation

Multiple perspectives = more trustworthy explanations

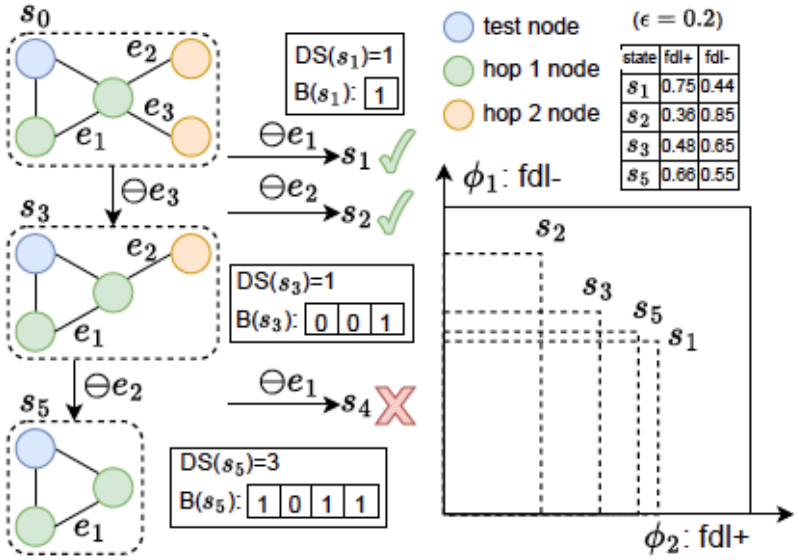
Interpretation is inherently multi-objective

Dazhuo Qiu, Haolai Che, Arijit Khan, and Yinghui Wu, "Interpreting Graph Inference with Skyline Explanations", in Proc. of IEEE International Conference on Data Engineering 2026.

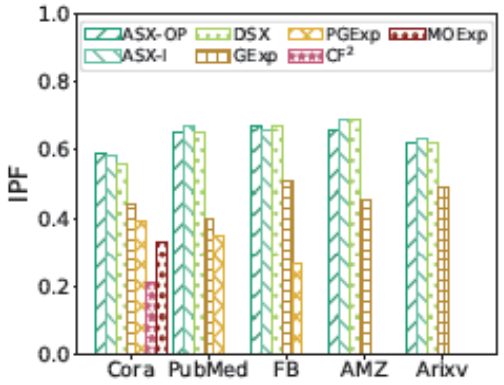
Generation: Onion Peeling + Pareto Filtering

- Start from L-hop neighborhood of the target node
- Iteratively remove outer edges first (onion peeling)
- Keep candidates that improve skyline dominance

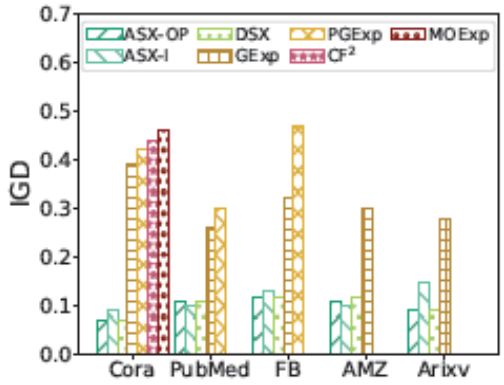
- Provable approximation guarantees for skyline quality
- Early pruning via dominance relations
- Parallel skyline explanations for large graphs



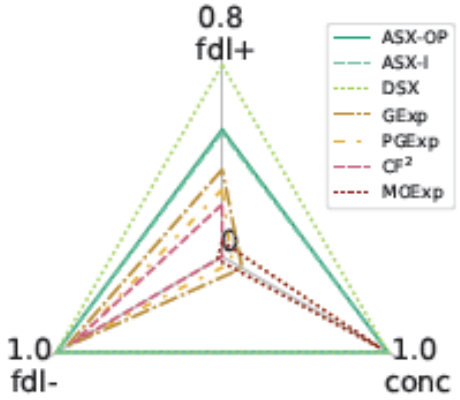
Evaluation: Effectiveness



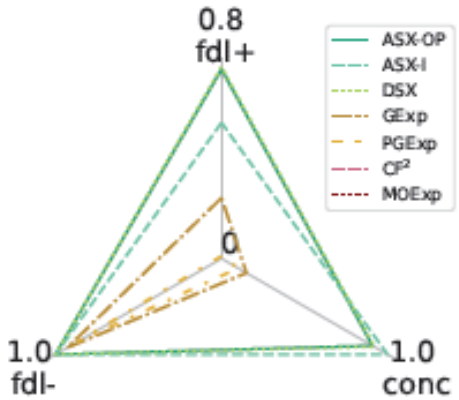
(a) IPF Overall Score



(b) IGD Overall Score



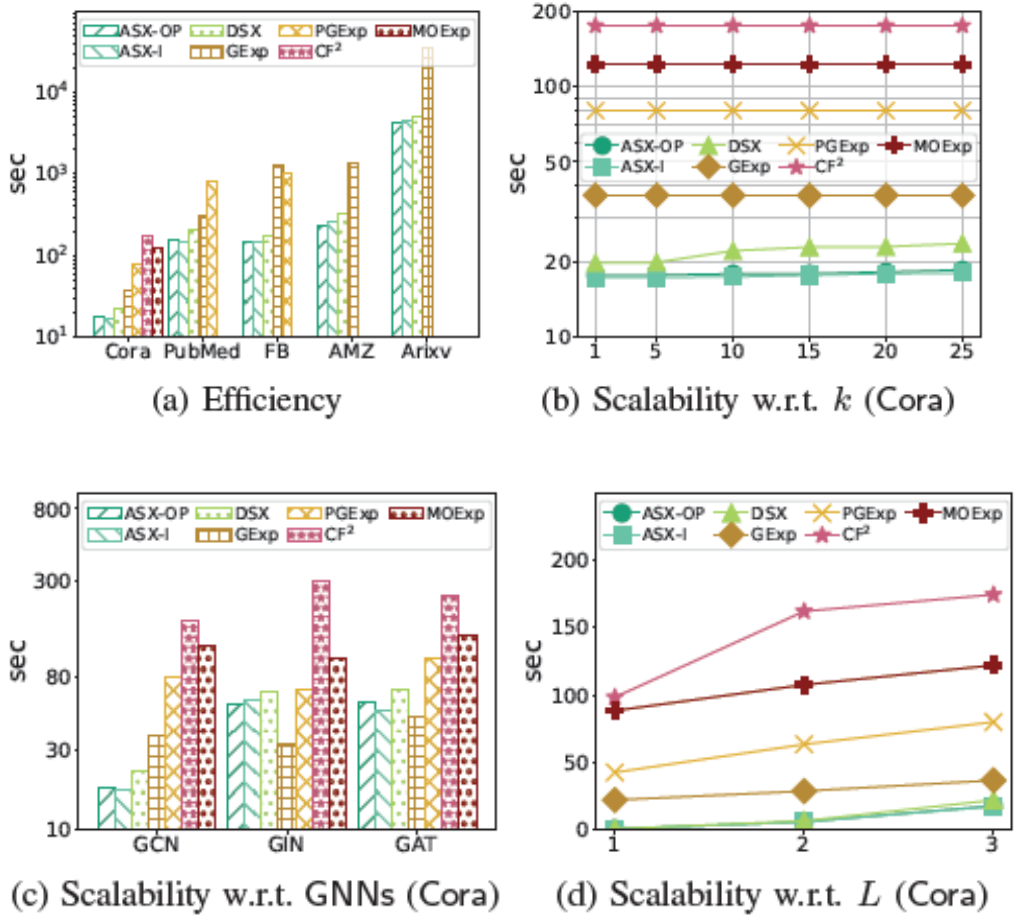
(c) MS on Cora



(d) MS on PubMed

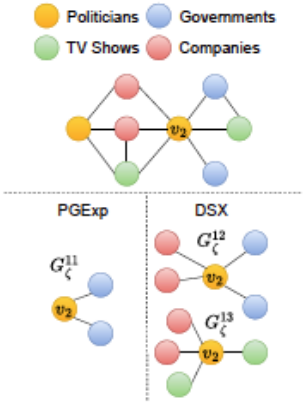
Overall effectiveness of our skyline explanation vs. SOTA explainers (GExp, PGExp, CF^2, MOExp).

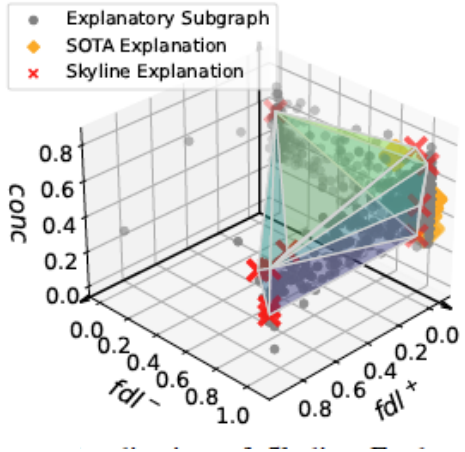
Evaluation: Efficiency

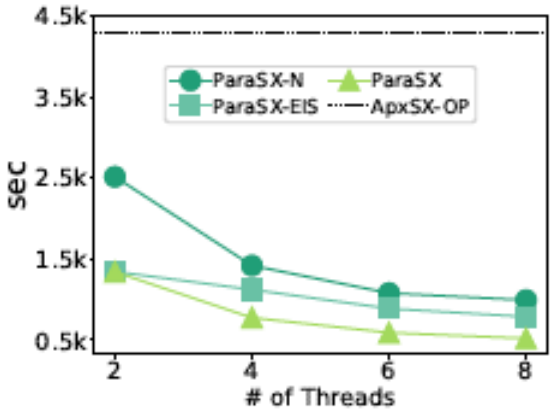


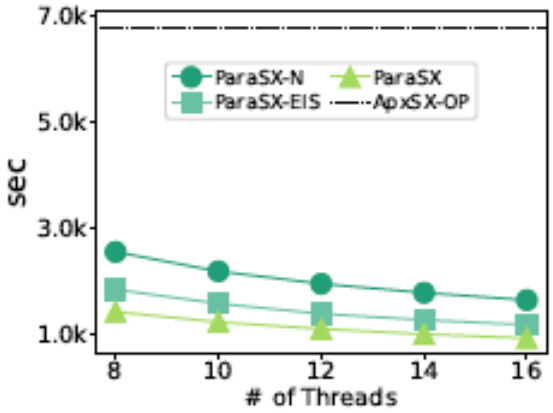
Efficiency and scalability of our skyline explanation vs. SOTA explainers (GExp, PGExp, CF^2, MOExp).

Scalability and Case Analysis









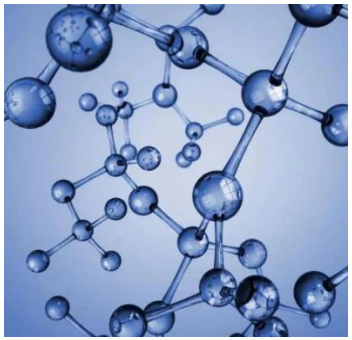
Scaling to million-scale and billion-scale graphs

Declarative Explanation

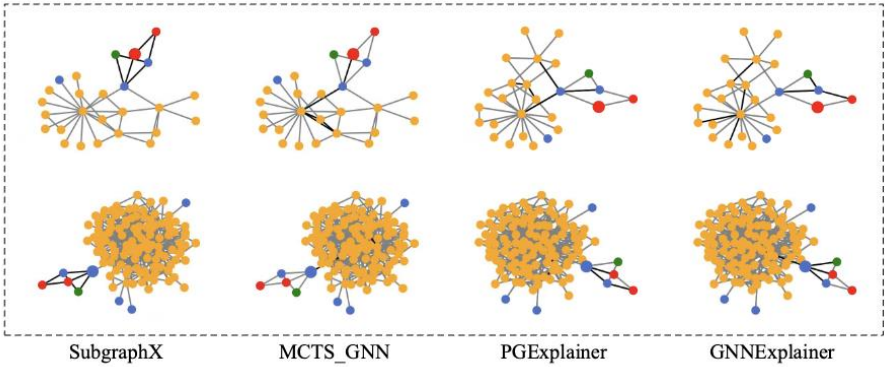
Xiangyu Ke

Declarative Explanatory Queries

Graph Neural Networks excel at learning graph structures through message passing and feature aggregation, seeing widespread use in social networks, bioinformatics, and recommendation systems.



Recent work therefore shifts toward higher-level explanation units, highlighting a common need: users do not just want "an explanation," but want to ask for explanations with specific targets, constraints, granularity, and purposes.



Declarative Explanatory Queries address this by providing a query-style interface—where users specify what they want to know and the system determines how to generate the corresponding explanation, enabling interactive, controllable.

Declarative Explanatory Queries

- **Benefits:**
  - **Task alignment & usefulness:** Explanations match the user's objective — debugging, auditing, regulatory reporting, or hypothesis testing — so they're directly actionable.
  - **Reproducibility & auditability:** Explicit constraints and query parameters
  - **Interactivity & composability:** Queries can be iteratively refined
  - **Resource-aware generation:** Backends can honor resource or latency budgets stated in the query
  - **Comparability & aggregation:** The same query language enables fair comparisons across models, datasets, or classes and supports aggregate analyses

- **Challenges:**
  - **Specification design:** Creating a language that is expressive yet usable is hard.
  - **Objective trade-offs:** Queries will often encode conflicting goals
  - **Optimization complexity:** Many useful query constraints lead to combinatorial search problems that are NP-hard in general.
  - **Model and data variability:** Stochastic models, randomized inference, or distribution shift can make query results unstable
  - **Verification & guarantees:** Users need confidence—how do we certify that a returned explanation satisfies the declared constraints, or how well it approximates the ideal objective?

Existing Declarative methods

---- Can I Trust the Explainer? Verifying Post-hoc Explanatory Methods [arXiv 19]

Motivation: Existing explainers are often compared without a clear "ground truth," making it impossible to know if we can truly trust their outputs on complex black-box models.

M: IF *"very good"* IN *input*: RETURN 0.9;
IF *"nice"* IN *input*: RETURN 0.7;
IF *"good"* IN *input*: RETURN 0.6;
RETURN 0.

x_1: "The movie was good, it was actually nice."
$M(x_1) = 0.7$

Feature-additivity	*Feature-selection*
nice: 0.4	{*nice*}
good: 0.3	
rest of tokens: 0	

x_2: "The movie was nice, in fact, it was very good."
$M(x_2) = 0.9$

Feature-additivity	*Feature-selection*
good: 0.417	{*good*, *very*}
nice: 0.367	
very: 0.116	
rest of tokens: 0	

The Declarative Approach： The paper formalizes explainability by ***declaring specific perspectives*** (Feature-additivity vs. Feature-selection) and building a verification framework based on provable neural architectures. By designing models with explicit logical constraints (e.g., the Handshake mechanism), the "correct" explanation is declared by design, allowing for the first time a rigorous audit of post-hoc explainers.

IF *"very good"* IN *input*: SELECT *"very"* & RETURN 1;
IF *"not good"* IN *input*: SELECT *"not"* & RETURN 0.1;
IF *"good"* IN *input*: SELECT *"good"* & RETURN 0.8;
ELSE SELECT ∅ & RETURN 0.5.

Figure 2: Example of handshake.

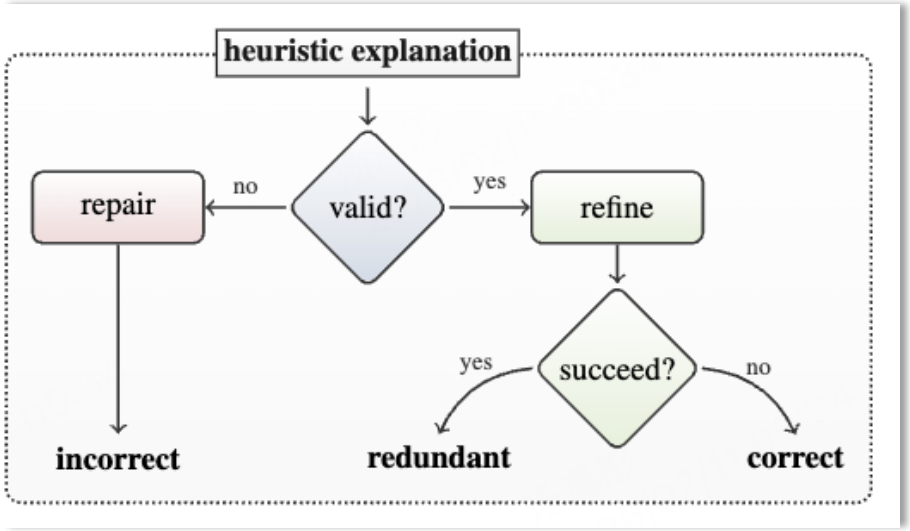
Figure 2 illustrates the handshake problem: non-selected tokens may still influence the prediction via information encoded in the selected tokens, making feature-selection explanations unreliable.

Existing Declarative method

---- Towards Trustable Explainable AI [AAAI 20]

Motivation: Existing explainers are heuristic: they rely on local sampling and "guess" which features are important. They offer no guarantee that the explanation is valid across the entire model space, leading to a "Trust Gap."

The Declarative Approach： This work treats explainability as a formal verification task. Instead of sampling, we declare what a valid explanation must be: a minimal set of features that logically guarantees the model's prediction. By encoding the ML model into logical constraints (SAT/SMT), the system can prove that the explanation is correct.



The framework acts as a Reasoning Oracle that audits existing explanations:

- ***Verify:*** *It checks if a given explanation holds true for all possible inputs.*
- ***Repair:*** *If an explanation is proven "invalid," the system uses logical reasoning to fix it.*
- ***Refine:*** *If an explanation is "redundant," the system strips away unnecessary features to find the prime implicant (PI)—the most concise, provably correct reason.*

In this context, **Declarative XAI** means defining an explanation by its logical requirements (what it must prove) rather than its computational heuristic (how it was sampled).

Overview of SliceGXQ [in submission]

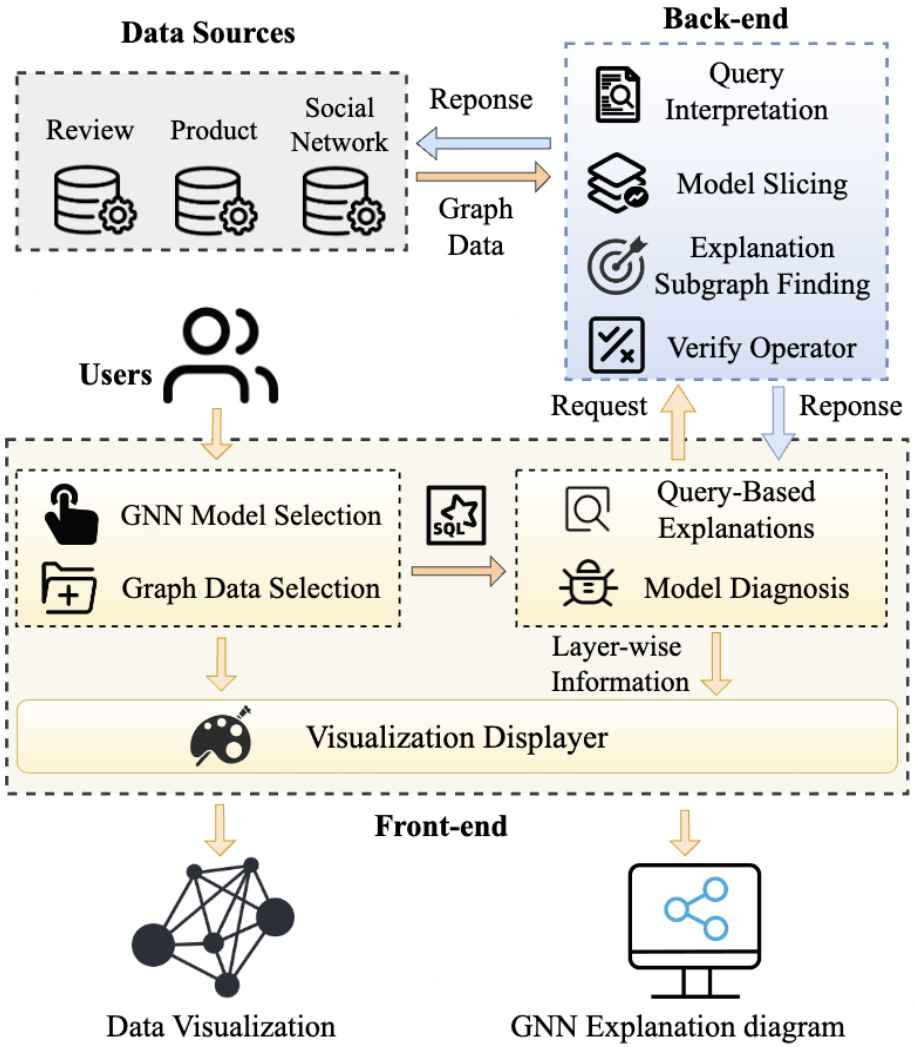
SliceGXQ turns SPARQL-style queries into layer-wise GNN explanations. It interprets the query, slices the model, finds explanatory subgraphs, and verifies the results before visualizing them.

Inputs

Users select a GNN model and a graph dataset from different data sources.

Outputs

SliceGXQ returns query-based explanations and model diagnosis with layer-wise information.



Back-end

The system performs query interpretation, then applies model slicing to focus on relevant layers/components, finds explanatory subgraphs, and verifies the explanation results.

Query-based Interaction

Users submit a SPARQL-style request to specify what explanations or diagnostics they need.

EXPLAIN $(\mathcal{M}, V_T)$ **AT** l_t
FROM $\{G, Ł\}$
WHERE P, k **WITH** *mode*

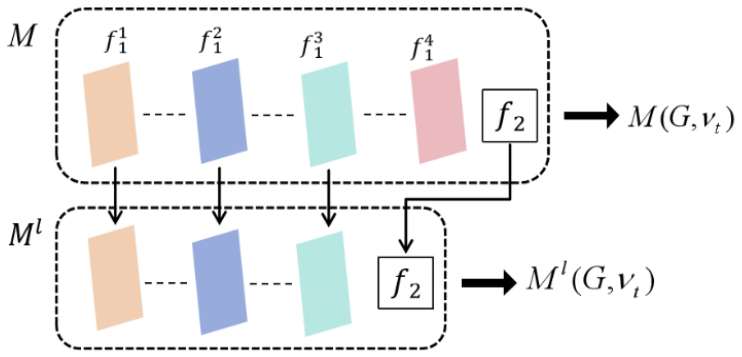
Core algorithms in SliceGXQ

Query-Driven Interaction

The introduction of a lightweight SPARQL-style interface empowers users to customize their explanation needs, making complex GNN internals accessible through simple, flexible queries.

The SliceGXQ query follows **EXPLAIN(M, V_T) AT l_t FROM {G, L} WHERE P, k WITH mode**, where it specifies the target model and nodes, the layer to inspect, the dataset/layer scope, constraints (pattern/size), and the explanation mode.

EXPLAIN $(\mathcal{M}, V_T)$ **AT** l_t
FROM$\{G, Ł\}$
WHERE P, k **WITH** *mode*



Q_1	EXPLAIN (M, {v}) At 3 FROM {G, {1}} WHERE 4
Q_2	EXPLAIN (M, {v}) At 3 FROM {G, ϕ} WITH diagnose
Q_3	EXPLAIN (M, {v_1, v_2}) At 3 FROM {G, ϕ} WITH interpret
Q_4	EXPLAIN (M, {v}) At 3 FROM {G, {1, 2}} WHERE <'*Fraud*','–','*Fraud*'>
Q_5	EXPLAIN (M, {v_1, v_2}) At 3 FROM {G, {1, 2}} ORDER BY node.id ASC

Some SPARQL-style query examples

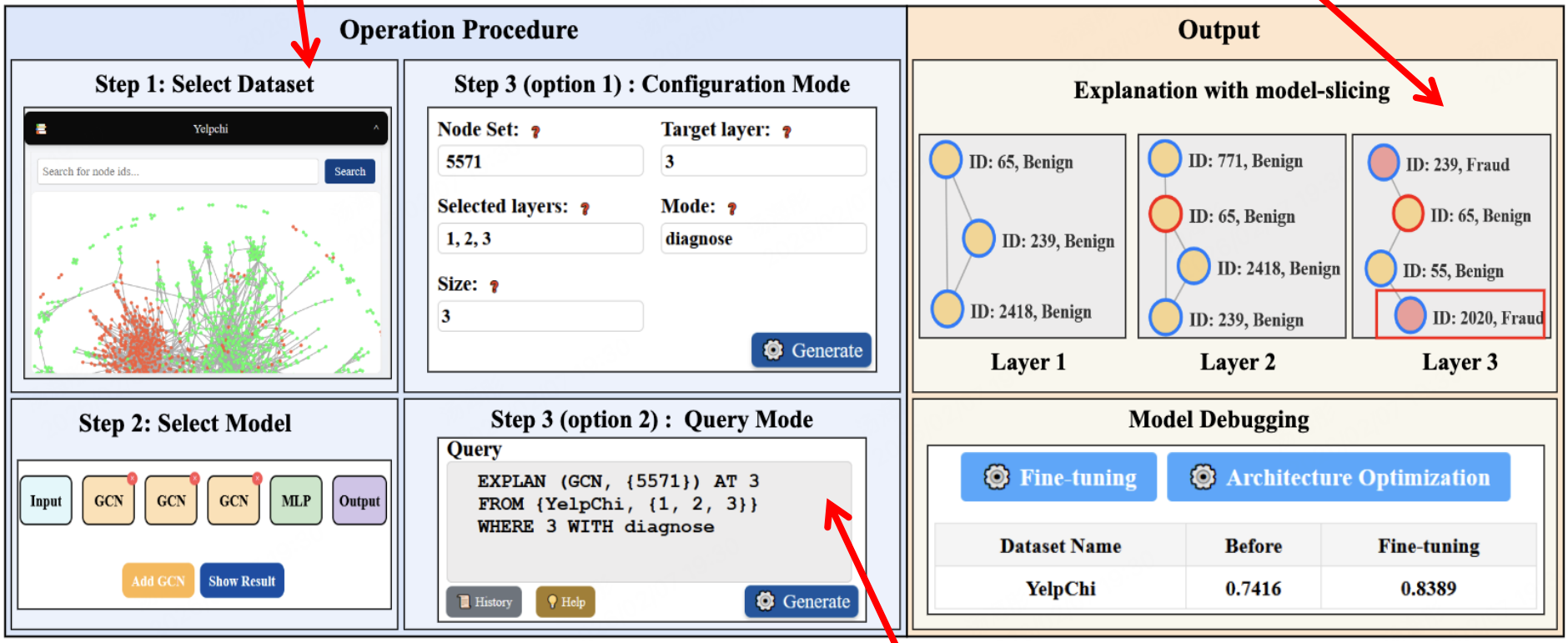
Layer-wise Insight

Unlike traditional "black-box" explainers, SliceGXQ provides deep, layer-specific insights, allowing users to trace how decision-making evolves across different GNN layers.

SliceGXQ Demonstration

Step 1 & 2 (Setup): Users first select a target graph dataset (e.g., YelpChi) and configure the GNN architecture (e.g., adding GCN layers) to be explained.

Layer-wise Explanations: The system outputs explanatory subgraphs for each layer (Layer 1–3), visualizing how the model's reasoning evolves and identifying critical features (e.g., distinguishing Fraud vs. Benign nodes).



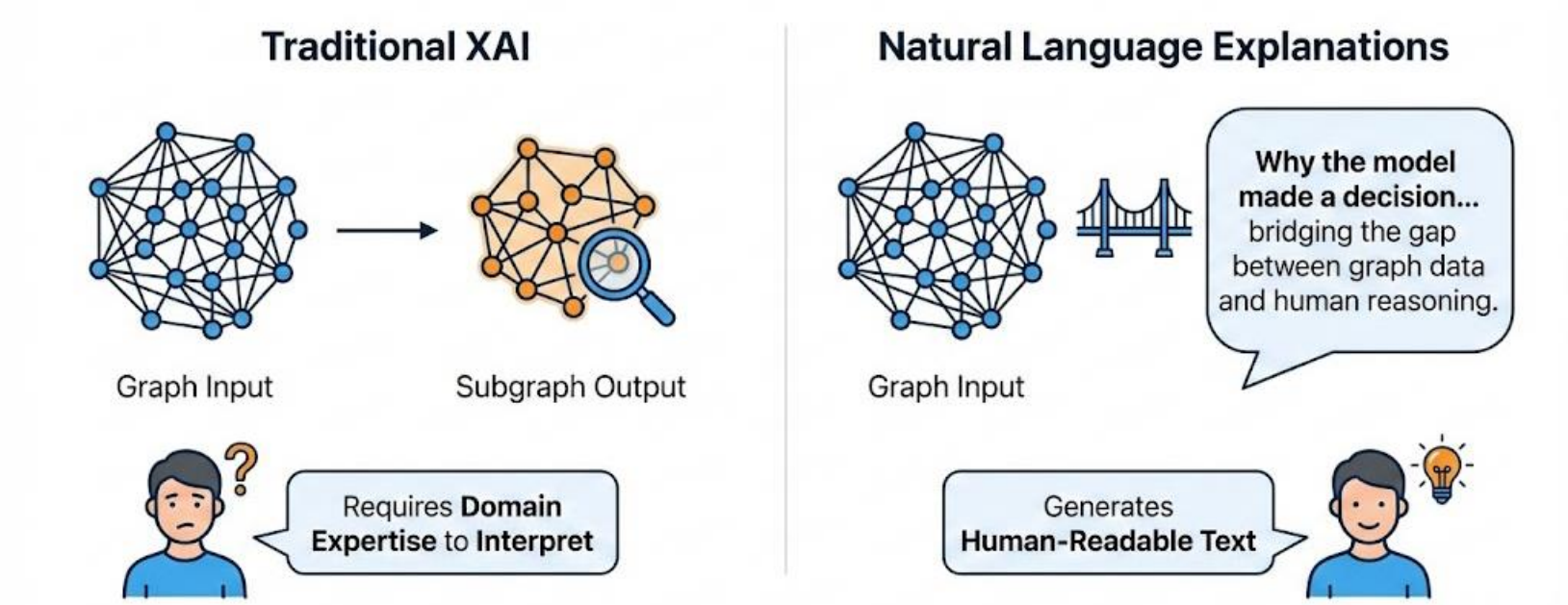
Step 3 (Request): Users can either use Configuration Mode for direct parameter settings or Query Mode to issue flexible SPARQL-style queries (e.g., EXPLAN (GCN, {5571}) AT 3...), specifying the target node, layer, and diagnostic mode.

Natural Language Explanation

Xiangyu Ke

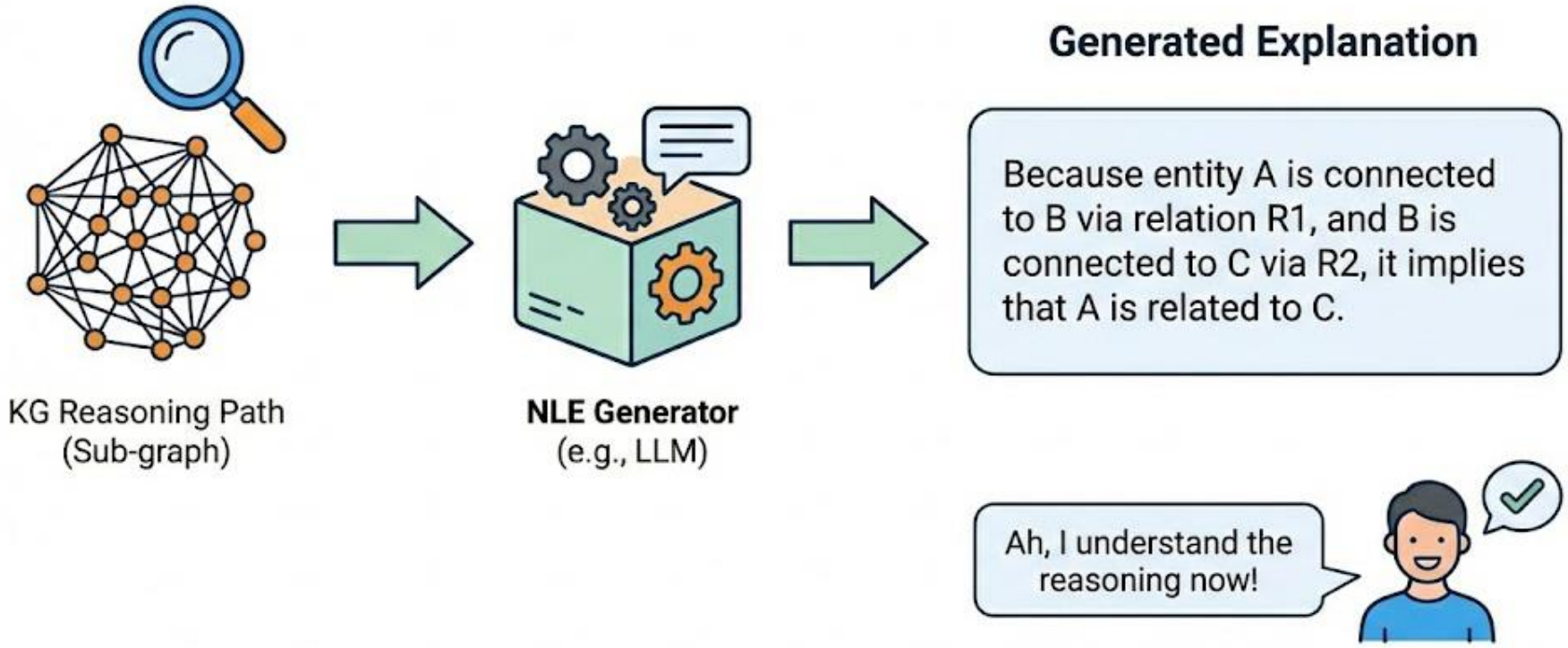
From Structural Subgraphs to Semantic Narratives

- **Definition:**
  - **Traditional XAI:** Outputs a subgraph (e.g., a set of nodes/edges) which requires domain expertise to interpret.
  - **Natural Language Explanations:** Generates human-readable text that describes why the model made a decision, bridging the gap between graph data and human reasoning.



Data-centric → **User-centric**: Shifting focus from "what the model sees" (pixels/nodes) to "what the user understands" (language)

Why Natural Language Explanation is Vital?



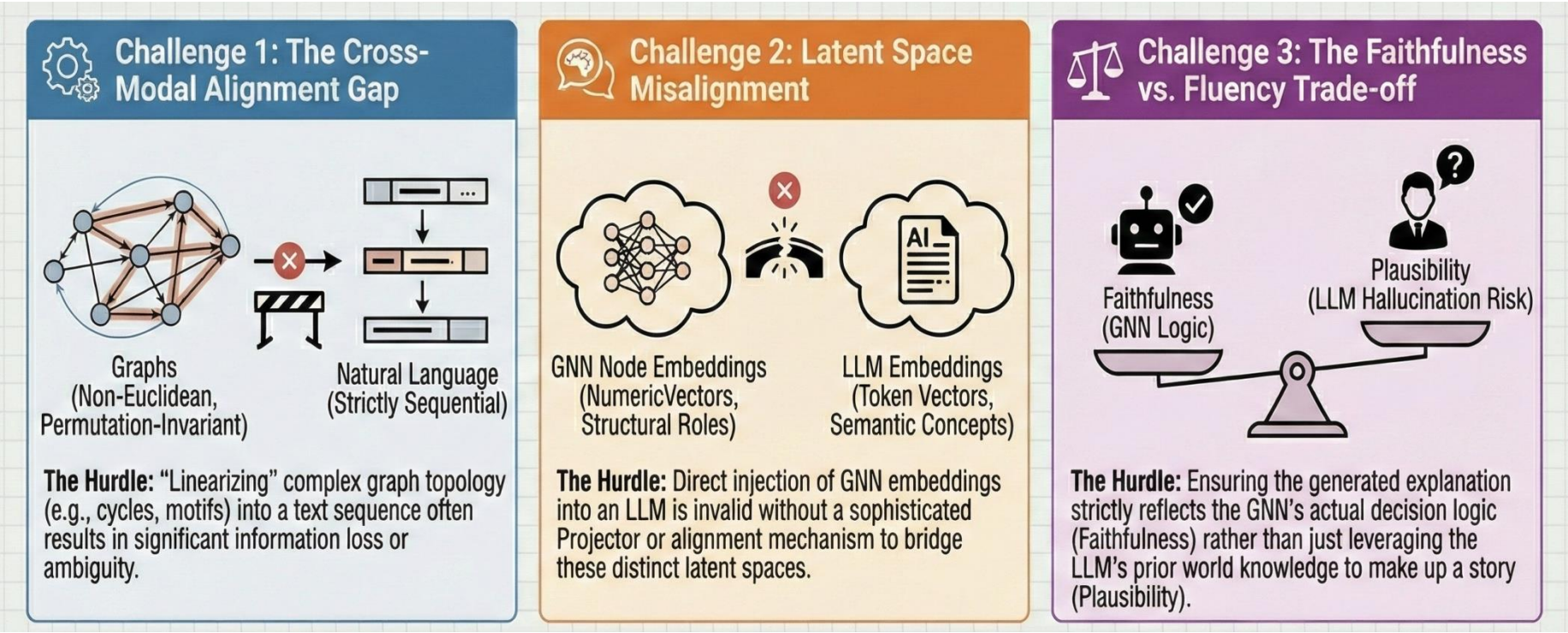
From Binary to Narrative: Transitioning from providing raw subgraphs to generating human-readable reasoning.

Semantic Enrichment: Leveraging the vast world knowledge of LLMs to describe complex graph patterns in natural language.

Interactive Transparency: Enabling non-experts to understand model decisions through logical dialogue instead of technical masks.

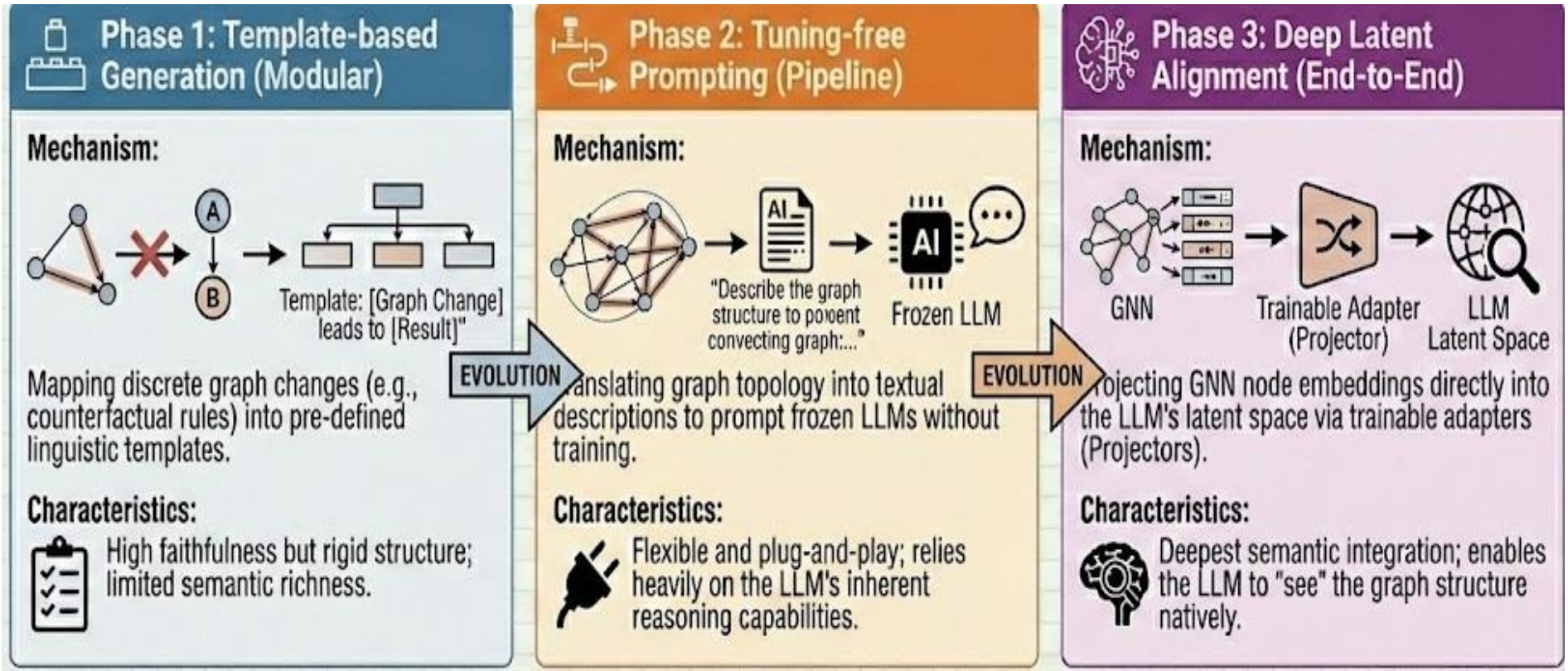
Why it is hard?

- **Challenges:**
  - (1) **Linearizing** complex **graph topology** into sequential text causes significant **information loss** and **ambiguity**.
  - (2) **Aligning** numerical **GNN embeddings** with semantic **LLM vectors** requires sophisticated **projection mechanisms**.
  - (3) Ensuring explanations strictly reflect **GNN logic**, avoiding **hallucinations** driven by LLM **prior knowledge**.

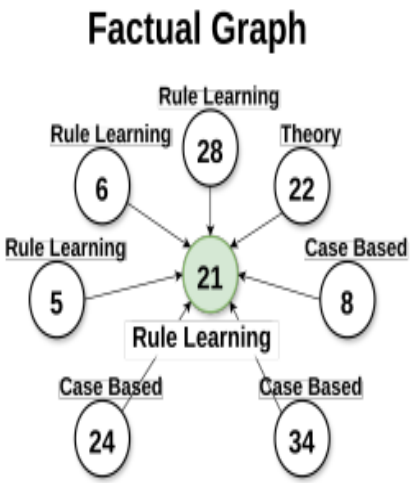


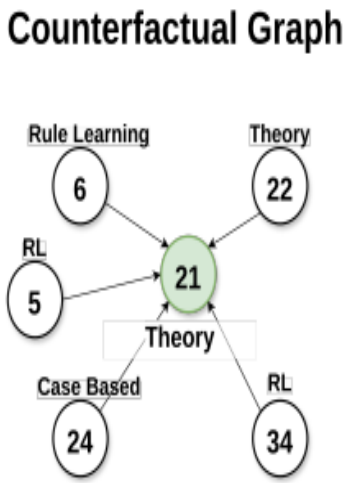
Evolution of NL-XGNN: From Templates to Deep Alignment

- **Phase 1: Template-based Generation (Modular).**
- **Phase 2: Tuning-free Prompting (Pipeline).**
- **Phase 3: Deep Latent Alignment (End-to-End).**



Natural Language Explanations





Given the factual graph: ***$factual graph description$*** and given the counterfactual example: ***$counterfactual graph description$*** and given the knowledge base about the dataset: ***$dataset knowledge$***, fill the dictionary and provide an explanation about the change in classification for the target node, please evaluate also the influences of neighbors nodes.

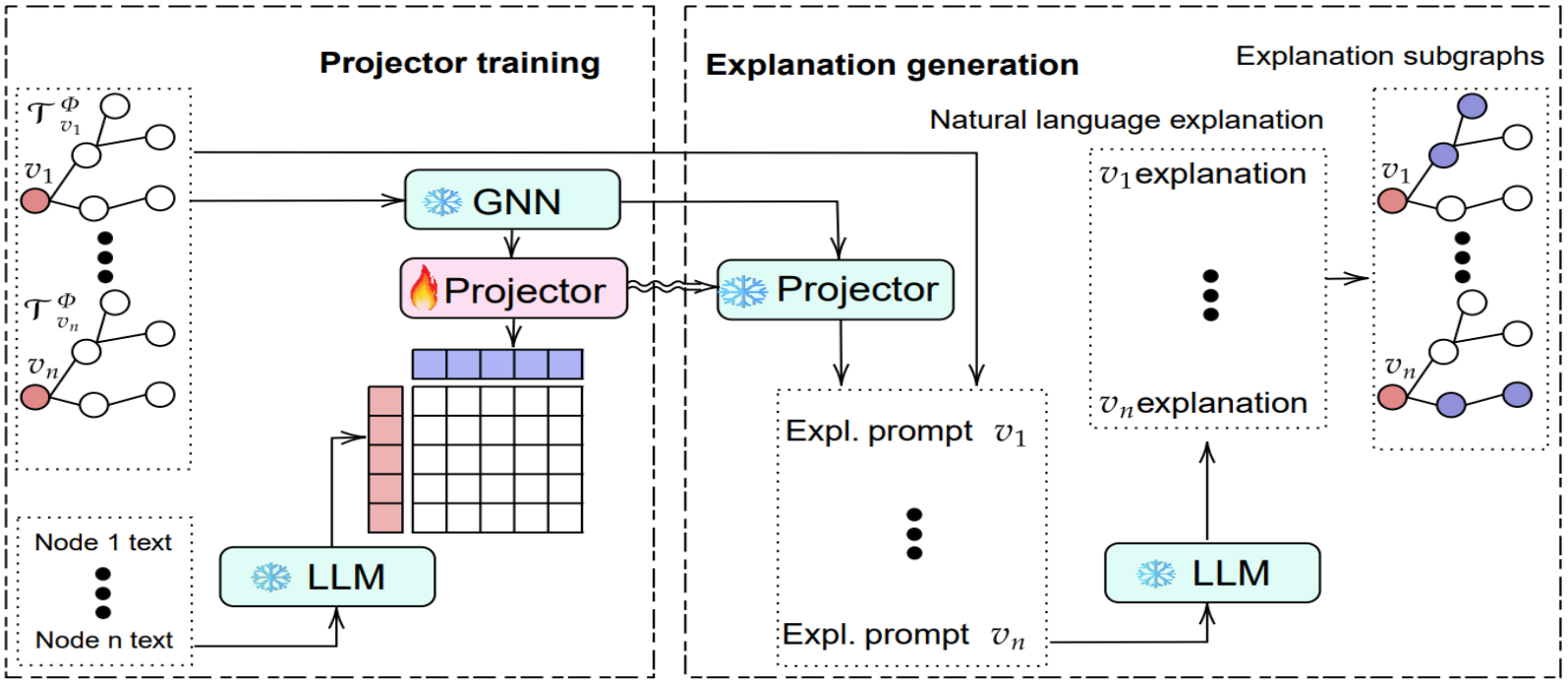
Evidence-to-Language Translation: Converting technical counterfactual instances into human-readable narratives.

Modular Integration: A plug-and-play approach compatible with any state-of-the-art counterfactual explainer

User-Centric Refinement: Utilizing LLMs to bridge the gap between abstract graph changes and domain-specific reasoning.

[arXiv 24] Giorgi, F., Campagnano, C., Silvestri, F., & Tolomei, G. **Natural language counterfactual explanations for graphs using large language models.**

Natural Language Explanations



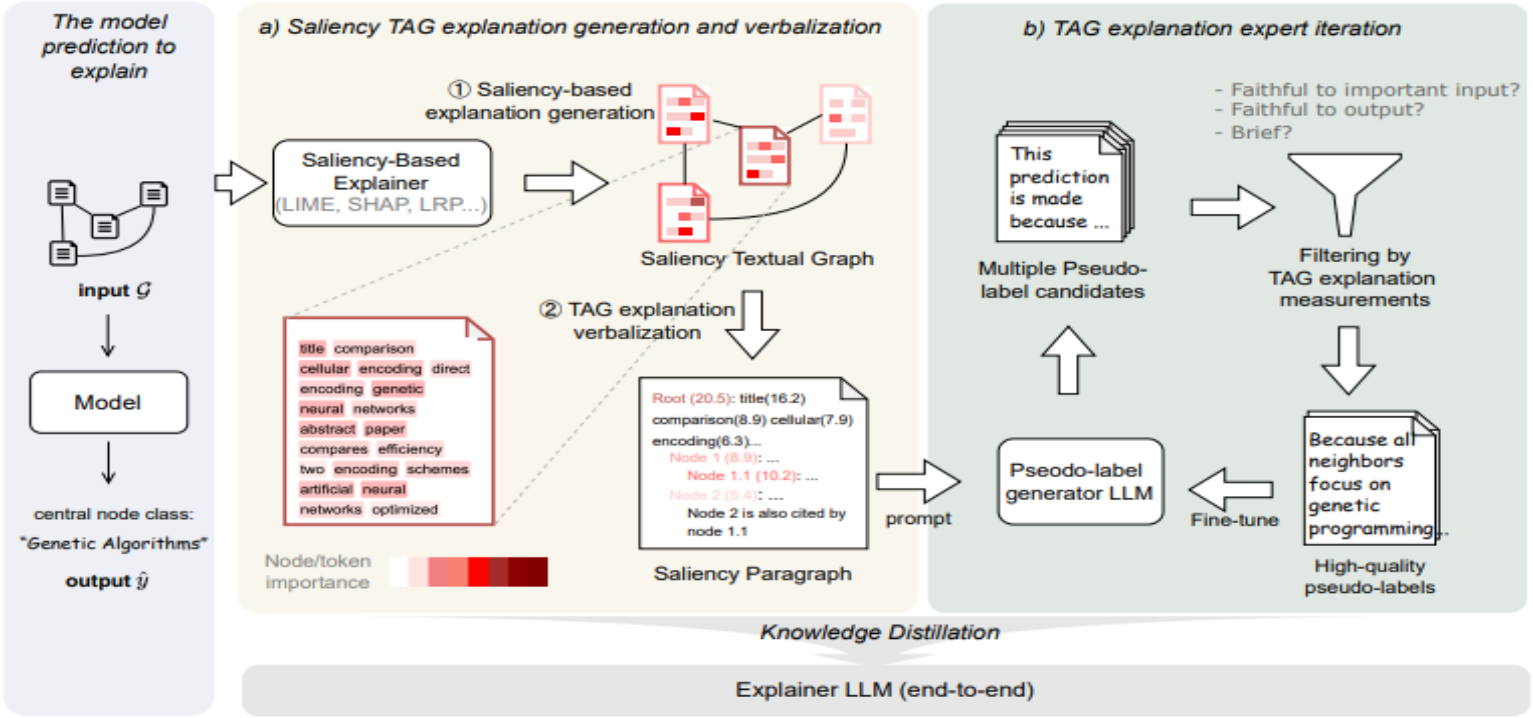
Embedding Projection: Projecting GNN node embeddings directly into the LLM's embedding space for deep alignment.

Hybrid Narrative Prompting: Interleaving "soft prompts" (GNN internal states) with "hard prompts" (textual graph structure).

Joint Subgraph & Rationale Generation: Simultaneously producing a concise explanation subgraph and a detailed natural language rationale.

[arXiv 25] Baghershahi, P., Fournier, G., Nyati, P., & Medya. **From Nodes to Narratives: Explaining Graph Neural Networks with LLMs and Graph Context.**

Natural Language Explanations



Saliency-to-Text Mapping: Converting technical saliency scores into natural language reasoning.

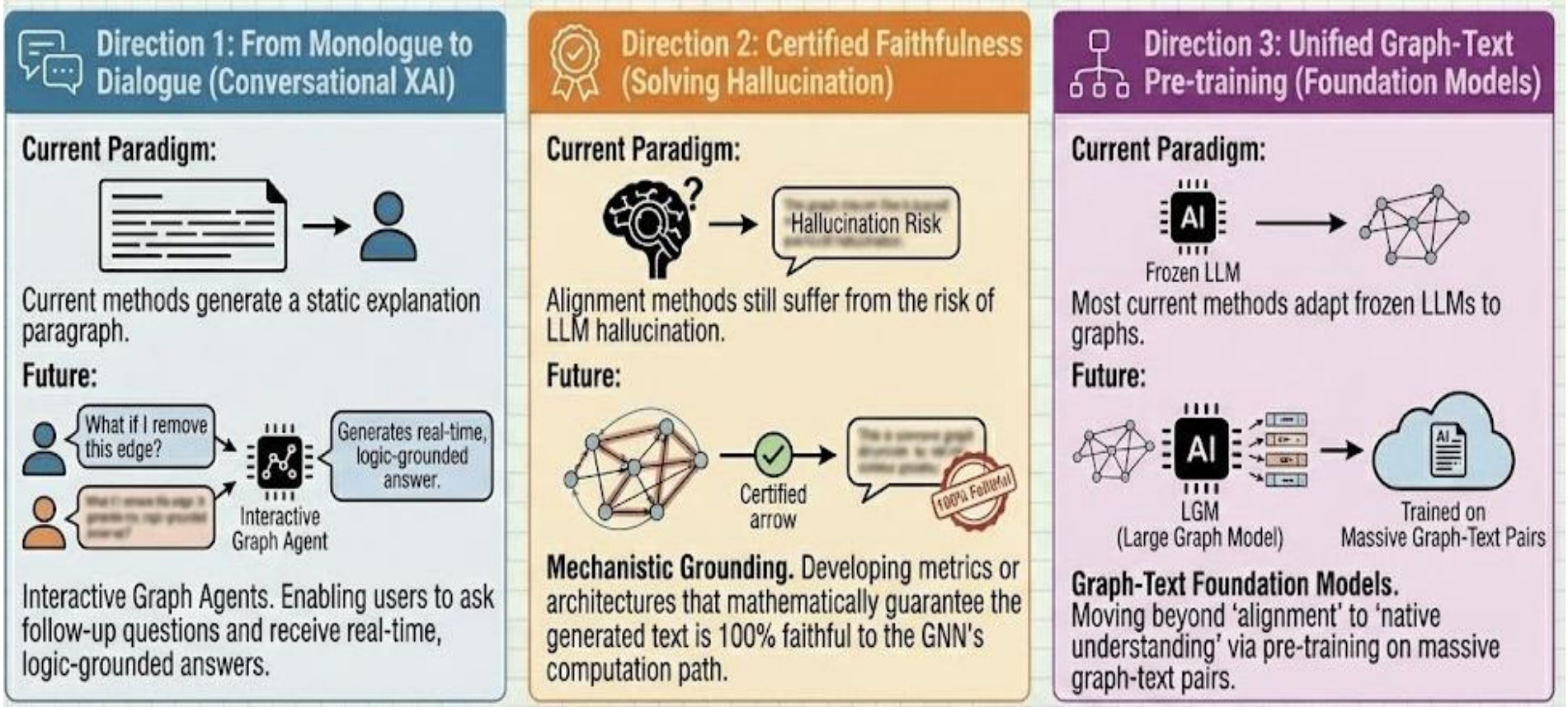
Expert Iteration Loop: Refining LLM outputs through iterative pseudo-label generation to ensure faithfulness.

Explanation Consistency: Aligning generated narratives with the underlying GNN's decision logic.

[ACL 25] Pan, B., Xiong, Z., Wu, G., Zhang, Z., Zhang, Y., Hu, Y., & Zhao, L. **Graphnarrator: Generating textual explanations for graph neural networks.**

Future Directions in Natural Language Explanations

- **Direction 1: From Monologue to Dialogue (Conversational XAI).**
- **Direction 2: Certified Faithfulness (Solving Hallucination)**
- **Direction 3: Unified Graph-Text Pre-training (Foundation Models)**



Counterfactual Explanation

Arijit Khan

Finding Counterfactual Evidences for Node Classification

Node classification with Graph Neural Networks (GNNs):

- Predicting labels for nodes in a graph by leveraging both node features and graph structure.

Counterfactual learning:

- The possibility of an alternative outcome if some of the premises were different from what were in reality ("counter to the facts").

GNN's limitations on high-stakes application domains:

- Lack interpretability and tend to inherit dataset biases, causing discriminatory decisions on sensitive attributes.

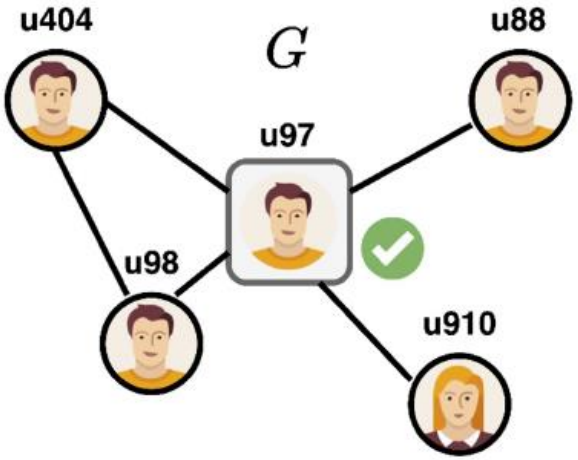
Dazhuo Qiu, Jinwen Chen, Arijit Khan, Yan Zhao, and Francesco Bonchi, "Finding Counterfactual Evidences for Node Classification", KDD 2025

Paper: https://arxiv.org/abs/2505.11396

Codes: https://github.com/DazhuoQ/Counterfactual_Evidence

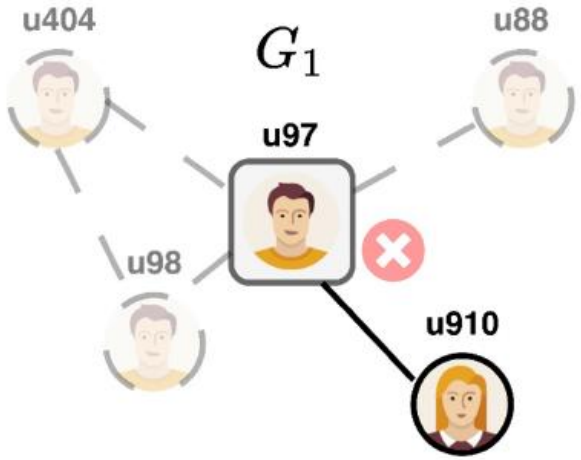
Counterfactual Evidence vs. Counterfactual Explanation

Input Graph of u97



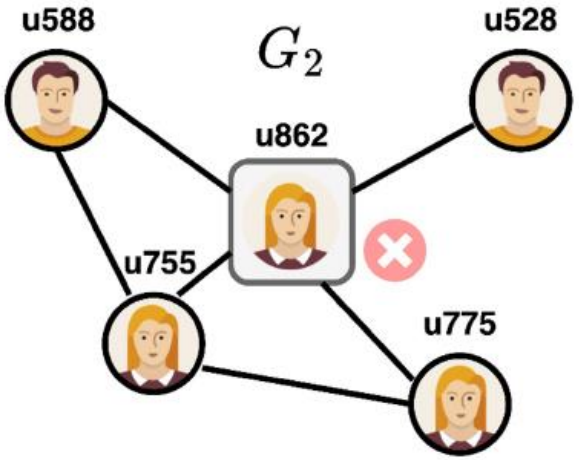
User	Gender	Age	Loan Duration (Month)	Loan Amount (Deutsch Mark)
u97	***Male***	***30-40***	***12-24***	***2k-3k***
u98	Male	30-40	36-48	2k-3k
u404	Male	40-50	12-24	2k-3k
u88	Male	30-40	12-24	3k-4k
u910	Female	30-40	36-48	4k-5k

Counterfactual Explanation



User	Gender	Age	Loan Duration (Month)	Loan Amount (Deutsch Mark)
u97	***Male***	***30-40***	***12-24***	***2k-3k***
u98	Male	30-40	36-48	2k-3k
u404	Male	40-50	12-24	2k-3k
u88	Male	30-40	12-24	3k-4k
u910	Female	30-40	36-48	4k-5k

Counterfactual Evidence



User	Gender	Age	Loan Duration (Month)	Loan Amount (Deutsch Mark)
u862	***Female***	***30-40***	***24-36***	***2k-3k***
u755	Female	30-40	24-36	1k-2k
u588	Male	40-50	12-24	1k-2k
u528	Male	30-40	36-48	2k-3k
u775	Female	18-30	24-36	1k-2k

Counterfactual Evidence

The need for counterfactual evidences:

- Avoid perturbing the dataset, as it may produce infeasible or unrealistic counterfactuals based on unobservable data.

Problem definition:

Given a node v from the test node set, another node u is called its counterfactual evidence (CE) if the following two conditions hold:

- v and u are assigned different labels by model M: $M(v) \neq M(u)$;
- L-hop neighbor subgraphs of v and u have a high similarity score, e.g., Kernel-based Similarity Score or $KS(v, u)$.

Kernel-based Similarity Search

Kernel Computation: $$\mathbf{x}_v^{l+1} = \alpha \cdot \mathbf{x}_v^l + \frac{1-\alpha}{|N(v)|} \sum_{u \in N(v)} \text{COSINE}(\mathbf{x}_v^l, \mathbf{x}_u^l) \cdot \mathbf{x}_u^l$$

Kernel-based Similarity: $$\text{KS}(v,u) = \frac{\mathbf{x}_v^{agg} \cdot \mathbf{x}_u^{agg}}{||\mathbf{x}_v^{agg}||_2 \cdot ||\mathbf{x}_u^{agg}||_2}$$

Top-1 Local & Global Counterfactual Evidences

Top-1 Local Counterfactual Evidence:

Given a query node v, the top-1 local CE is a node u that:

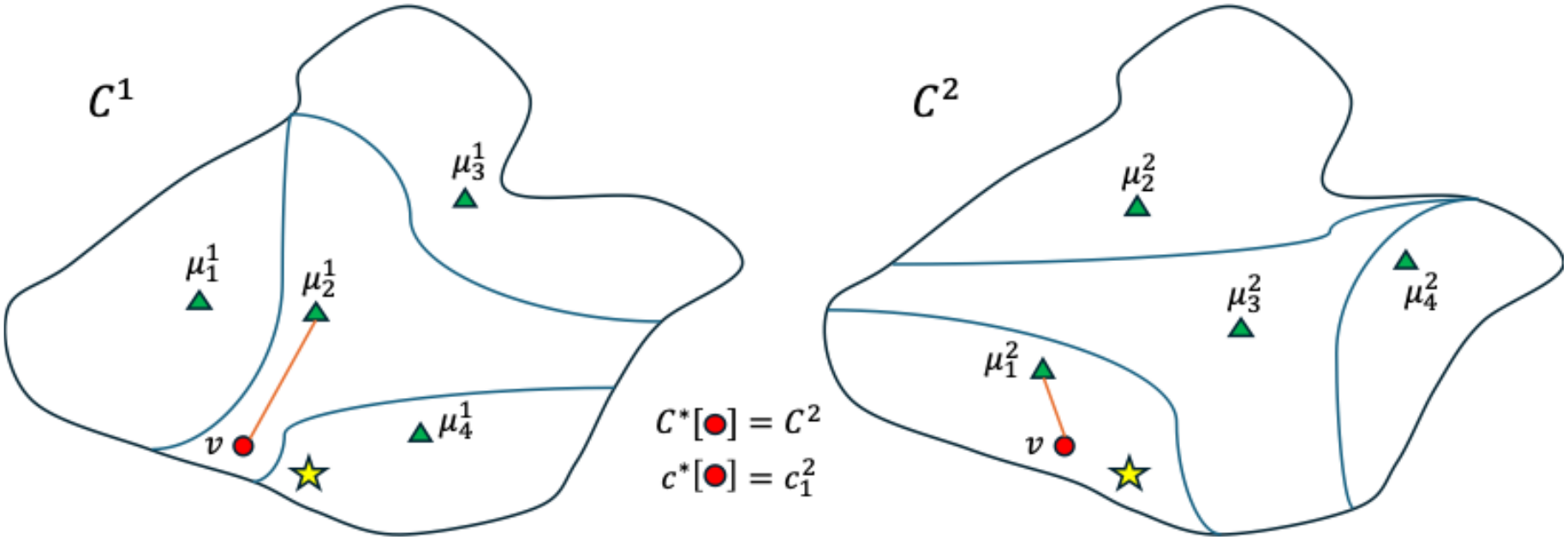
- has a different predicted label w.r.t. v;
- attains the highest similarity score $KS(v, u)$ compared to all other nodes in the test set.

Top-1 Global Counterfactual Evidence:

Given a test set, the top-1 global CE is a pair of nodes (v, u) such that:

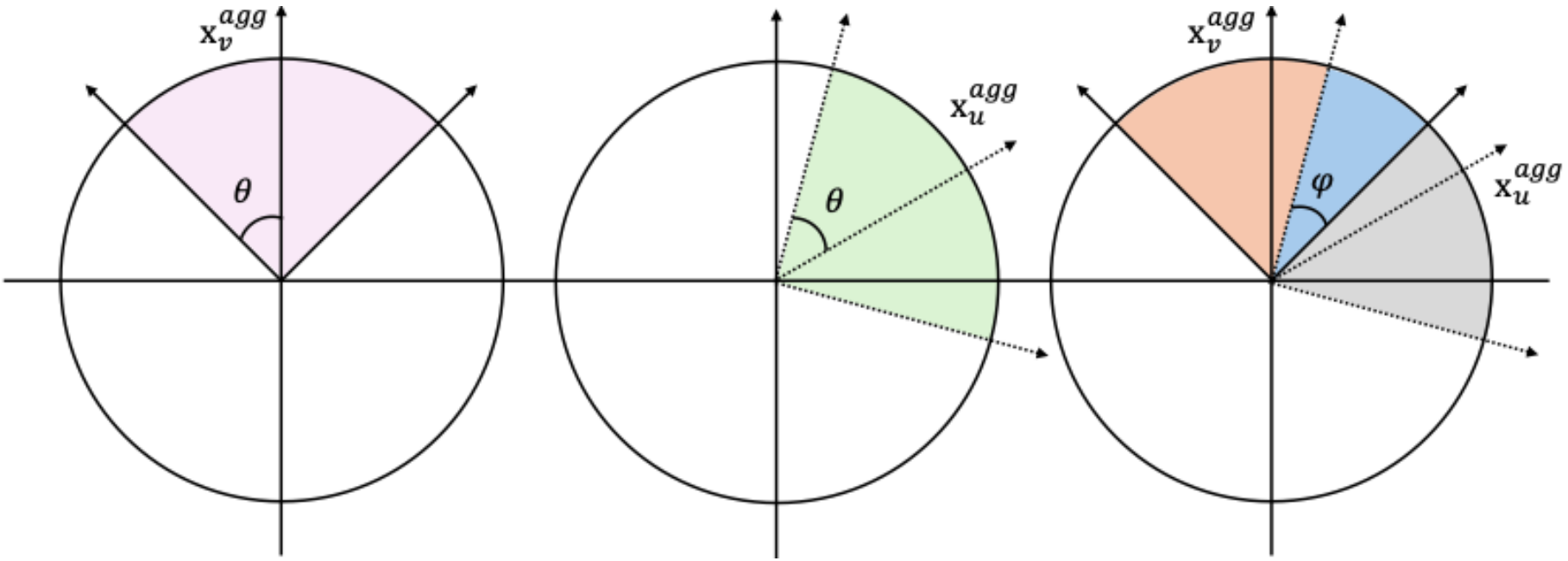
- v and u have different predicted labels;
- the pair has the highest similarity score $KS(v, u)$ among all possible node pairs in the test set.

Algorithm: Supplementary Partitioning



Two partitions C^1 and C^2, number of clusters in each partition m=4, green triangles are centroids of each cluster, the red node is the query node, the orange lines indicate the distance to the centroid and also reflect the weights, and the yellow star indicates the top-1 CE of the query node.

Algorithm: Weighted Clustering



Example of vector intersection in 2-dimension. Consider v as the cluster centroid, with all the possible similar vectors located within purple area. Suppose u is a vector assigned to another cluster. However, it is possible that u's top-1 CE falls inside of v's cluster—specifically, into the blue area. Therefore, we utilize this characteristic and treat the proportion of blue area as a weight when determining the cluster assignment for node u.

Algorithm: LocalCE-B&I

Algorithm 1 LocalCE-B

Input: Graph G, GNN M, test set V_{test}, query node v.
Output: Top-1 local counterfactual evidence $\mathsf{LCE}_{opt}(v)$ for v.

1: $\mathsf{LCE}_{opt}(v) :=$ None.
2: **for** $u \in V_{test} \backslash v$ **do**
3: **if** $M(v) \neq M(u)$ and $\mathsf{KS}(v,u) > \mathsf{KS}(v, \mathsf{LCE}_{opt}(v))$ **then**
4: $\mathsf{LCE}_{opt}(v) := u$.
5: **return** $\mathsf{LCE}_{opt}(v)$.

Algorithm 2 LocalCE-I

Input: Graph G, GNN M, test set V_{test}, query node v, set of partitions P.
Output: Top-1 local counterfactual evidence $\mathsf{LCE}_{opt}(v)$ for v.

1: $\mathsf{LCE}_{opt}(v) :=$ None.
2: Identify optimal partition $C^*[v]$.
3: Identify optimal cluster $c^*[v]$.
4: **for** $u \in c^*[v]$ **do**
5: **if** $M(v) \neq M(u)$ and $\mathsf{KS}(v,u) > \mathsf{KS}(v, \mathsf{LCE}_{opt}(v))$ **then**
6: $\mathsf{LCE}_{opt}(v) := u$.
7: **return** $\mathsf{LCE}_{opt}(v)$.

Algorithm: GlobalCE-B&I

Algorithm 3 GlobalCE-B&I

Input: Graph G, GNN M, test set V_{test}, set of partitions P.
Output: Top-1 global counterfactual evidence $\text{GCE}_{opt}(V_{test})$.

1: Select LocalCE-B or LocalCE-I for local CE identification.
2: Identify top-1 LCE for each test node based on the selected local algorithm.
3: Identify top-1 GCE among the top-1 LCEs.
4: **return** $\text{GCE}_{opt}(V_{test})$.

Experiments: Settings

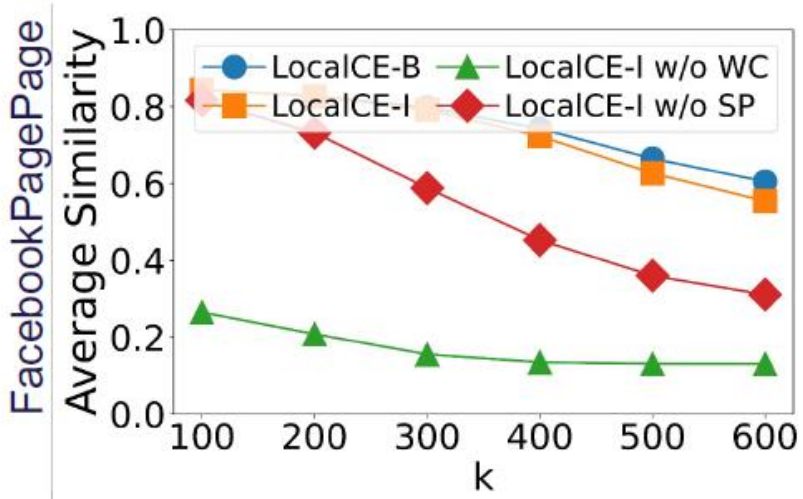
Statistics of datasets

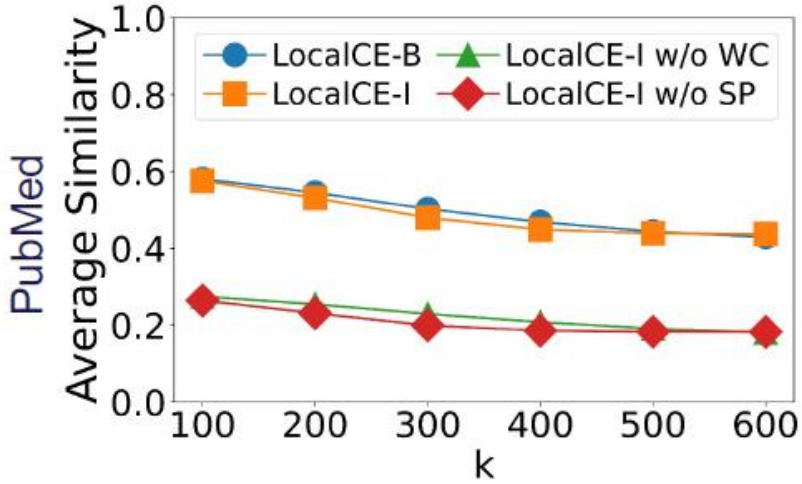
dataset	# nodes	# edges	# node features	# class labels
German	1,000	22,242	27	2
Bail	18,876	321,308	18	2
Cora	2,708	10,556	1,433	7
PubMed	19,717	88,648	500	3
FacebookPagePage	22,470	342,004	128	4
AmazonProducts	1,569,960	264,339,468	200	107

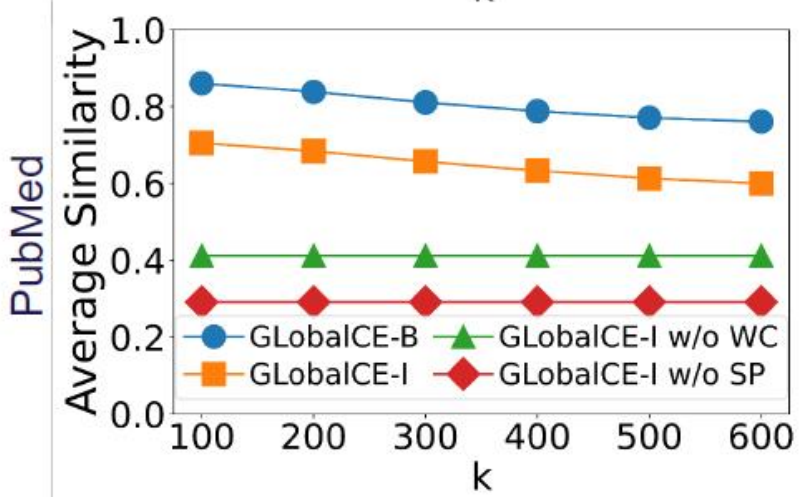
GNNs: GCN, GAT, GIN, MPNN, SAGE.

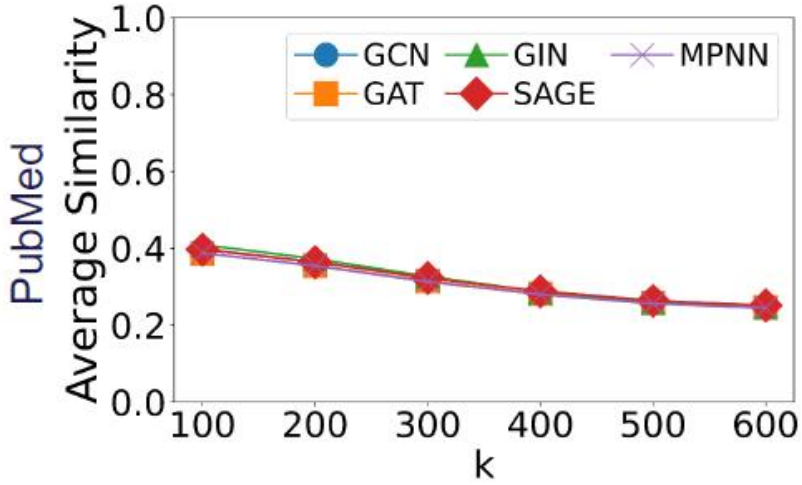
Competitors: LocalCE-I, LocalCE-B, LocalCE-I w/o WC, LocalCE-I w/o SP.

Experiments: Effectiveness & Generalizability

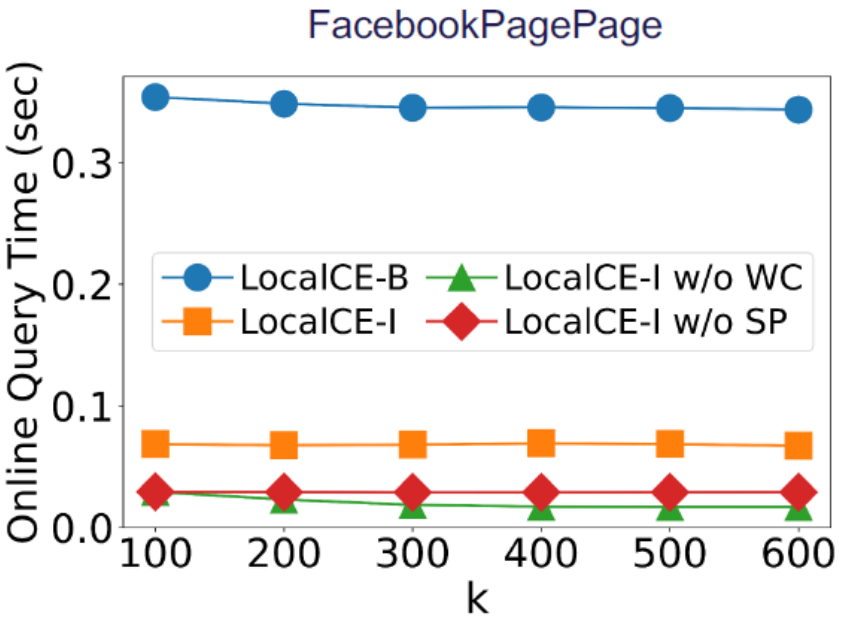


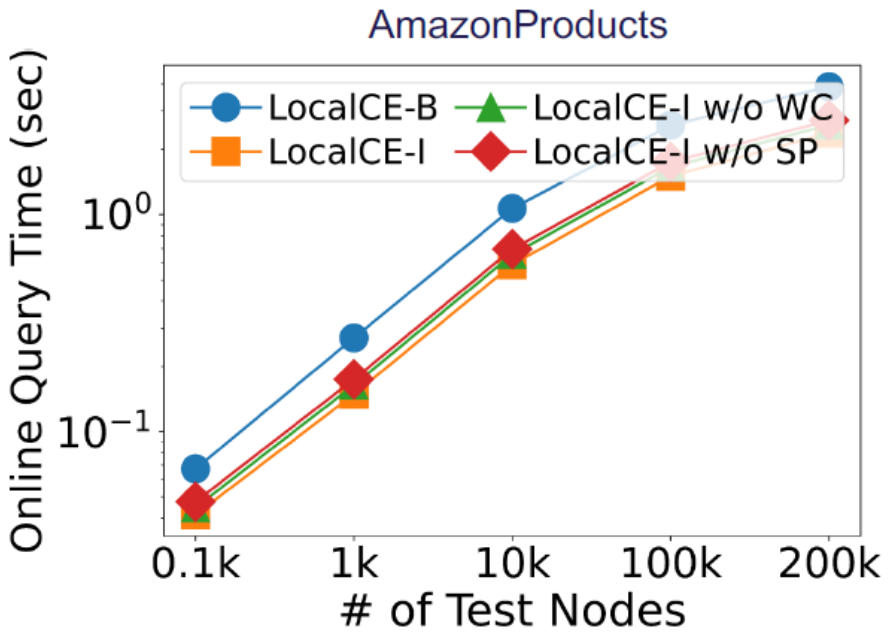






Experiments: Efficiency & Scalability





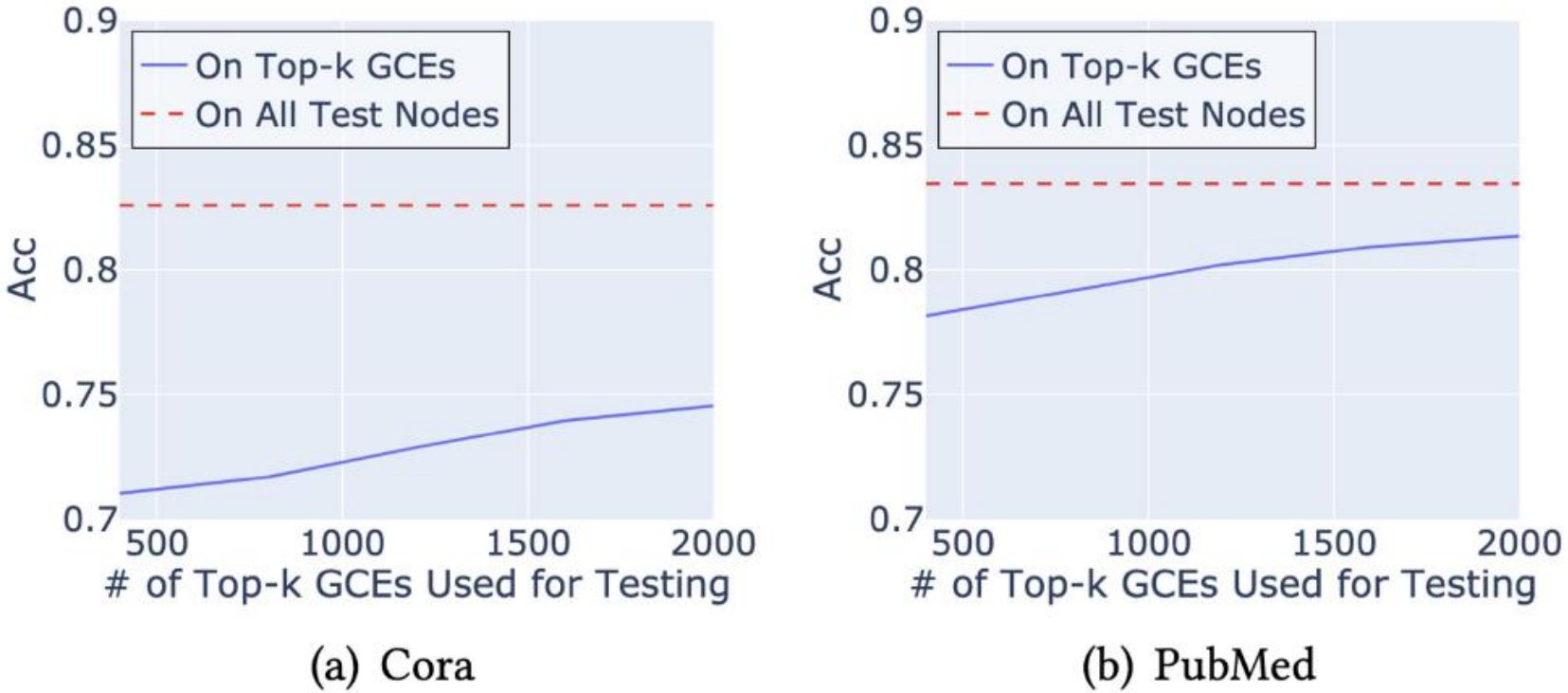
Applications: Revealing Unfairness of GNNs

Node features and their discrimination scores (DS) considering GCN [ICLR17] and FairGNN [WSDM21]: *Bail* dataset.

GCN	
Features	*DS*
WHITE	0.55
WORKREL	0.52
FILE3	0.52
SUPER	0.46
FILE1	0.41
MARRIED	0.37
FILE2	0.35
FELON	0.33
SCHOOL 8-13	0.32
PROPTY	0.27

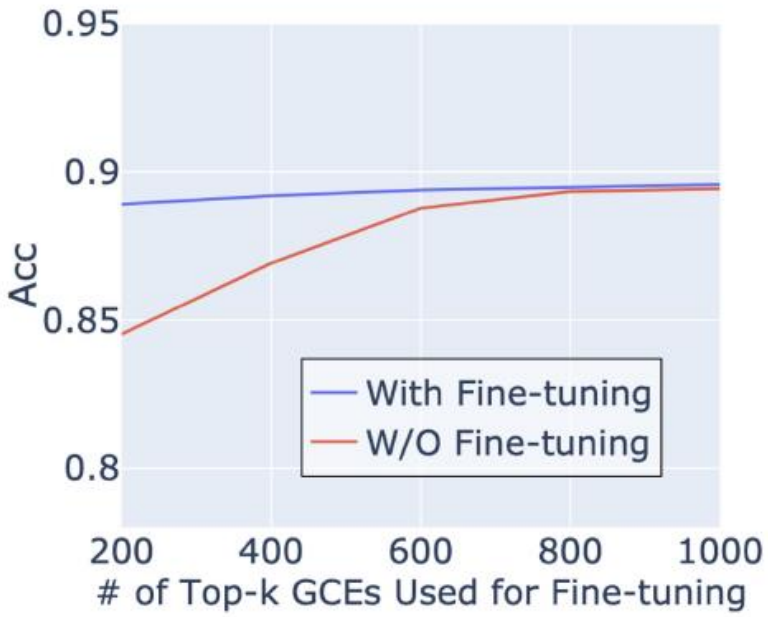
FairGNN	
Features	*DS*
TIME 0-30	0.79
TIME > 30	0.79
SUPER	0.52
FELON	0.51
FILE3	0.48
WHITE	0.48
WORKREL	0.48
MARRIED	0.46
FILE2	0.44
PROPTY	0.42

Applications: Verifying Prediction Errors

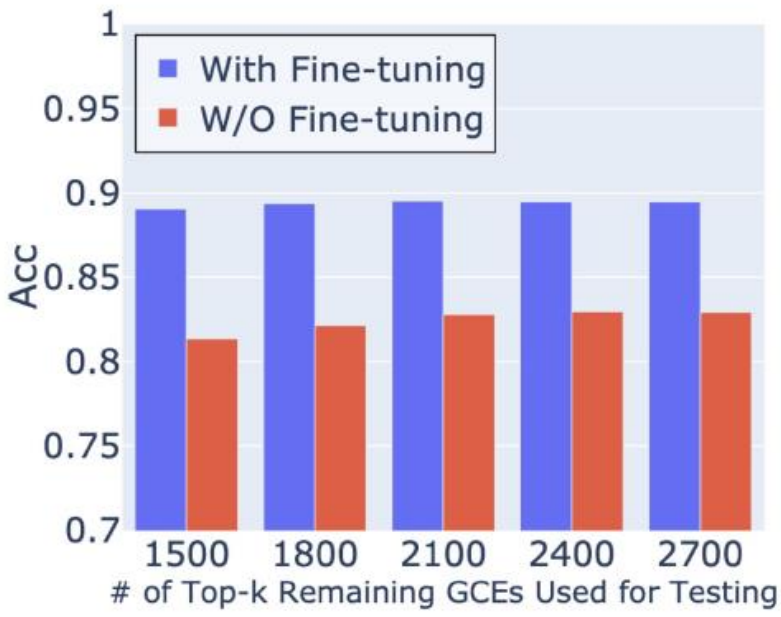


Accuracy within the top-k Global Counterfactual Evidences (GCEs): Accuracy within the GCEs is significantly lower for smaller k, mainly consisting of borderline nodes, which are difficult for the GCN to classify correctly.

Applications: Fine-tuning with CE



(a) PubMed



(b) PubMed

(a): Fine-tuning the GCN model with a set of top-k Global Counterfactual Evidences (GCEs) as a validation set improves the accuracy on the remaining test nodes. **(b)**: Using top-1200 GCEs to fine-tune the GCN: 1) improves the accuracy on the remaining test nodes, 2) remains consistently high after fine-tuning on both borderline and relatively easier cases.

ATEX-CF: Attack-Informed Counterfactual Explanations for GNNs

GNNs excel at node classification but remain **opaque** in critical domains (healthcare, finance, scientific discovery).

Counterfactual explanations answer: "What must differ for a different outcome?"

→ Identify **minimal graph changes** that flip a model's prediction.

Problem: Existing counterfactual methods rely almost exclusively on **edge deletions (E^-)**.

→ They overlook **missing relations (E^+)** whose addition could substantially alter predictions.

Meanwhile: Adversarial attacks on GNNs demonstrate that adding just **2–5 edges** can flip node predictions!

Yu Zhang, Sean Bin Yang, Arijit Khan, Cuneyt Gurcan Akcora,
"ATEX-CF: Attack-Informed Counterfactual Explanations for Graph Neural Networks", ICLR 2026

Paper: https://arxiv.org/abs/2602.06240

Codes: https://github.com/zhangyuo/ATEX_CF

ATEX-CF: Attack-Informed Counterfactual Explanations for GNNs

Key Question: Can we unify adversarial attacks with counterfactual explanationsto produce better, more comprehensive explanations?

Counterfactual Explanations vs. Adversarial Attacks

	Counterfactual Expl.	Adversarial Attack
Goal	Explain prediction	Cause misclassification
Primary Operation	Edge deletion (E^-)	Edge addition (E^+)
Constraint	Minimal & plausible	Small budget κ
Shared Effect	*Both flip the GNN prediction*	

Counterfactual (CF):

Removes existing edges

Reveals which relations support the current prediction

Adversarial Attack:

Adds new edges

Reveals which missing relations could alter the outcome

⟹ **Complementary perspectives on the same graph!**

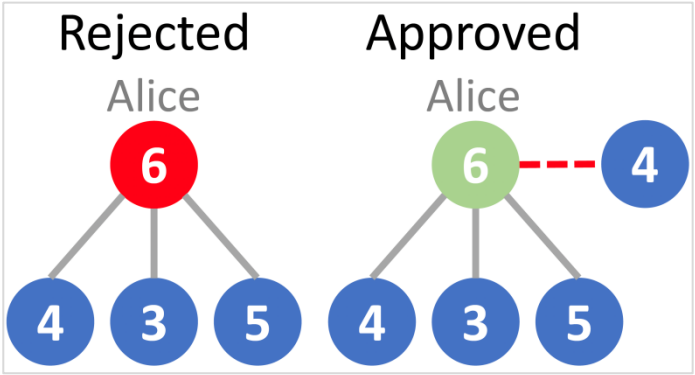
Case Study: Limitations of Deletion-Only Methods

❑ Loan Decision Dataset:

Loan approval requires: income > 5 AND degree > 3

Alice: income = 6 (✓), degree = 3 (✗)

Model predicts: **Rejected**



Deletion-based methods cannot help Alice; ATEX-CF finds a realistic edge addition.

❑ What happens?

Deletion-based CF: Removing edges further reduces degree → Fails!

Unconstrained addition: Link to billionaire → Succeeds but implausible

ATEX-CF: Identifies a feasible peer connection → Flips prediction & plausible!

Theoretical Foundation: Bridging Attacks & Explanations

Hypothesis: The edges added by a successful evasion attack overlap with the most influential edges in a counterfactual explanation.

Formal Statement

For target node v, let $\Delta G(E^+)$ be the attack-added edges that flip model f's prediction,and CFEx(G) the counterfactual explanation subgraph.
Then: Sim($\Delta G(E^+)$, CFEx(G)) is high

Implications:

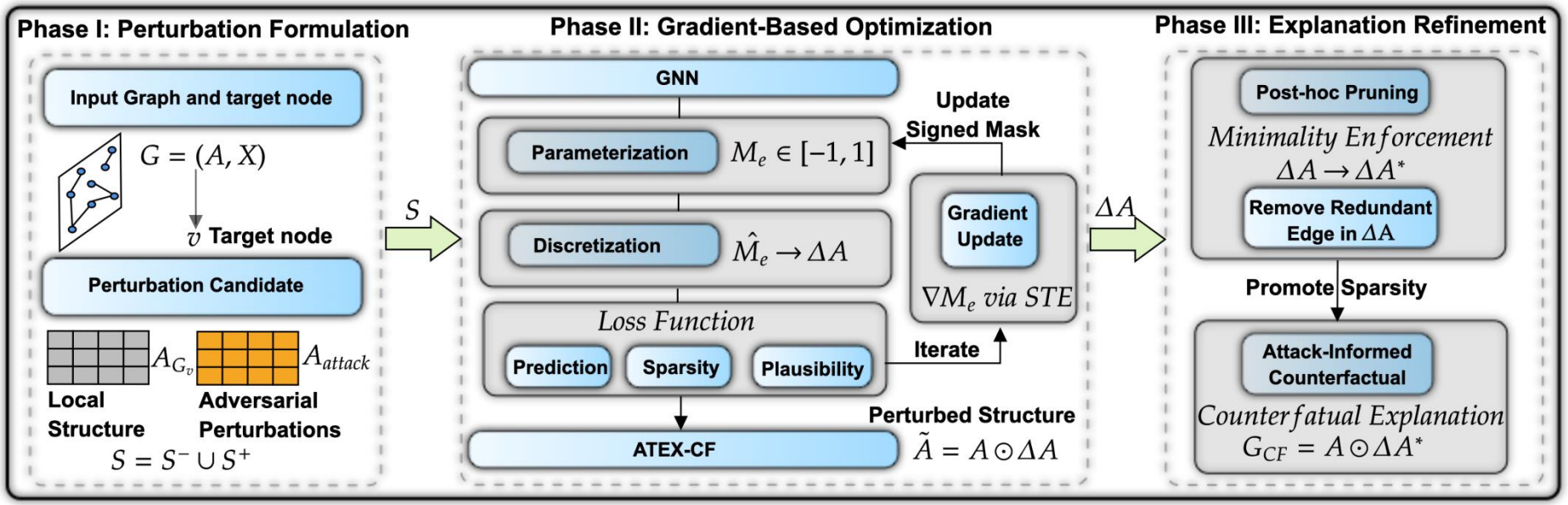
Attack-added edges are natural **counterfactual candidates**

Attack methods provide **efficient search** over the combinatorially large space of possible edge additions

Supported by gradient-based reasoning and empirical similarity measures

⟹ **This motivates integrating attack semantics into counterfactual generation.**

ATEX-CF: Framework Overview



End-to-end workflow: Original graph → Attack candidates → Candidate set $S = S^- \cup S^+$ → Joint optimization → Counterfactual graph

Joint Optimization: Three-Objective Loss

$$\min \; L(\Delta A) = \lambda_1 \cdot L_{pred}\,(\text{Impact}) + \lambda_2 \cdot L_{dist}\,(\text{Sparsity}) + \lambda_3 \cdot L_{plau}\,(\text{Plausibility})$$

Impact (L_{pred})

Push prediction away from original class until flip occurs.

Uses NLL loss with indicator function; Set zero once flipped.

Sparsity (L_{dist})

Minimize $\|\Delta A\|_0$: total number of edge edits.

Keeps modified graph close to original for interpretability.

Plausibility (L_{plau})

Penalizes unnatural changes:
$\alpha_{deg} \cdot \text{DegAnom}$
$+ \alpha_{motif} \cdot \text{MotifViol}$

Prevents implausible degree/clustering jumps.

Balances effectiveness (flip prediction) with realism (sparse & plausible edits)

Candidate Selection

❑ Candidate Set Construction:

- **S^- (Deletions):** Local structure — existing edges in (l+1)-hop neighborhood of target node v
- **S^+ (Additions):** Edges from adversarial attack (GOttack) — identifies influential nodes via graph orbit structure

$$S = S^- \cup S^+$$

- **Final subgraph of target node v:**

$$A_v = A_{G_v} \text{ (local)} + \Delta A_{attack} \text{ (adversarial)}$$

Signed-Mask Perturbation

- **Signed-Mask Perturbation:**

Each candidate edge e gets continuous mask

$M_e \in [-1, 1]$:

$M_e > 0$: addition

$M_e < 0$: deletion

$M_e \approx 0$: no change

Forward: Discretize to {-1, 0, +1} via thresholding and Top-κ edits for sparsification

Backward: Straight-Through Estimator (STE) enables gradient flow

Post-hoc pruning: Greedy removal of redundant edges to ensure minimality

ATEX-CF Algorithm

Algorithm 1: ATEX-CF: Counterfactual Generator

Require: Graph $G = (\mathbf{A}, X)$, model f, target node v, candidate set $\mathcal{S}$

Initialize mask $M_e \leftarrow \mathbf{0}$ over $\mathcal{S}$
for $t = 1$ to $T_{\max}$ **do**
  $\widehat{M}_e \leftarrow \textsc{Threshold}(M_e, \tau^+, \tau^-)$ ▷ Discretize
  $\Delta\mathbf{A} \leftarrow \textsc{Top-}\kappa(|M_e|)$ ▷ Sparsify
  Evaluate $\mathcal{L}(M_e)$ on $\mathbf{A} \odot \Delta\mathbf{A}$
  $M \leftarrow M - \eta \nabla_M \mathcal{L}(M)$ ▷ Update via STE
  if flipped(v) **and** $\|\Delta\mathbf{A}\|_0$ stable **then**
    break
  end if
end for
return $\textsc{Prune}(\Delta\mathbf{A}, G, f, v)$ ▷ See Alg. 2

❑ **Key Design Choices:**

SGD optimizer, lr = 0.001, 200 epochs max

Budget κ = 5 edges (default)

Symmetric thresholds $\tau^+ = \tau^- = 0.5$

Loss weights: λ_1=1.5, λ_2=0.5, λ_3=0.5

Early stopping on flip + stability

Post-hoc greedy pruning for minimality

Experimental Results: Meta-Ranking Across 6 Datasets

- **Datasets:** BA-Shapes, Tree-Cycles, Loan-Decision (synthetic); Cora, Chameleon, ogbn-Arxiv (real)
- **GNNs:** GCN, GAT, Graph Transformer | Budget: $\kappa = 5$

Method	Misclass.↓	Fidelity↓	ΔE ↓	Plausib.↓	Time↓	Avg Rank	Wins
CF-GNNExplainer	4.7	4.8	2.0	2.3	9.5	4.67	1
INDUCE	6.3	6.8	4.5	7.8	2.8	5.67	1
C2Explainer	5.0	6.2	5.7	5.8	8.7	6.27	1
GNNExplainer	6.8	7.5	4.7	5.5	3.3	5.57	1
PGExplainer	8.2	7.7	4.2	5.8	1.0	5.37	6
Nettack	3.3	2.5	8.8	8.0	4.5	5.43	2
GOttack	4.8	4.3	8.8	8.0	3.0	5.80	0
ATEX-CF (Ours)	**1.2**	**1.3**	**1.0**	**1.2**	**7.3**	**2.40**	**20**

Experimental Results: Meta-Ranking Across 6 Datasets

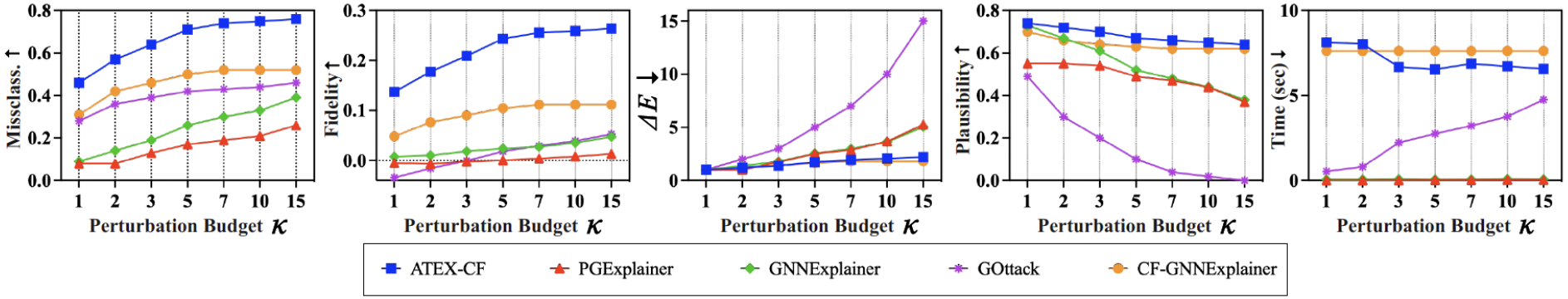
- **Datasets:** BA-Shapes, Tree-Cycles, Loan-Decision (synthetic); Cora, Chameleon, ogbn-Arxiv (real)
- **GNNs:** GCN, GAT, Graph Transformer | Budget: κ = 5

Method	Misclass.↓	Fidelity↓	ΔE ↓	Plausib.↓	Time↓	Avg Rank	Wins
CF-GNNExplainer	4.7	4.8	2.0	2.3	9.5	4.67	1
INDUCE	6.3	6.8	4.5	7.8	2.8	5.67	1
PGExplainer	8.2	7.7	4.2	5.8	1.0	5.37	6
Nettack	3.3	2.5	8.8	8.0	4.5	5.43	2
GOttack	4.8	4.3	8.8	8.0	3.0	5.80	0
ATEX-CF (Ours)	**1.2**	**1.3**	**1.0**	**1.2**	**7.3**	**2.40**	**20**

Average ranks (↓) across 6 datasets (lower = better); **Wins** out of 30 metric–dataset cells

⟹ ATEX-CF wins **20/30** cells; best overall rank **2.40** vs. 4.67 for the runner-up

Budget Sensitivity & Key Findings



Performance vs. perturbation budget κ on Cora (GCN)

❑ Key Observations:

- ATEX-CF consistently outperforms all baselines across budgets κ = 1 … 15
- Edit size of ATEX-CF grows mildly with κ; attack baselines must exhaust all allowed edits
- **Ablation:** Each loss component is essential — removing L_{dist} or L_{plau} degrades performance
- **Pruning** reduces edits (1.71→1.62) with no accuracy loss
- **Asymmetric costs:** Controllable trade-off between E^+ and E^- via cost weight C

Summary of Contributions

1. **Unified Perspective:** First theoretical bridge between adversarial attacks and counterfactual explanations in GNNs — attack edge additions serve as counterfactual candidates.

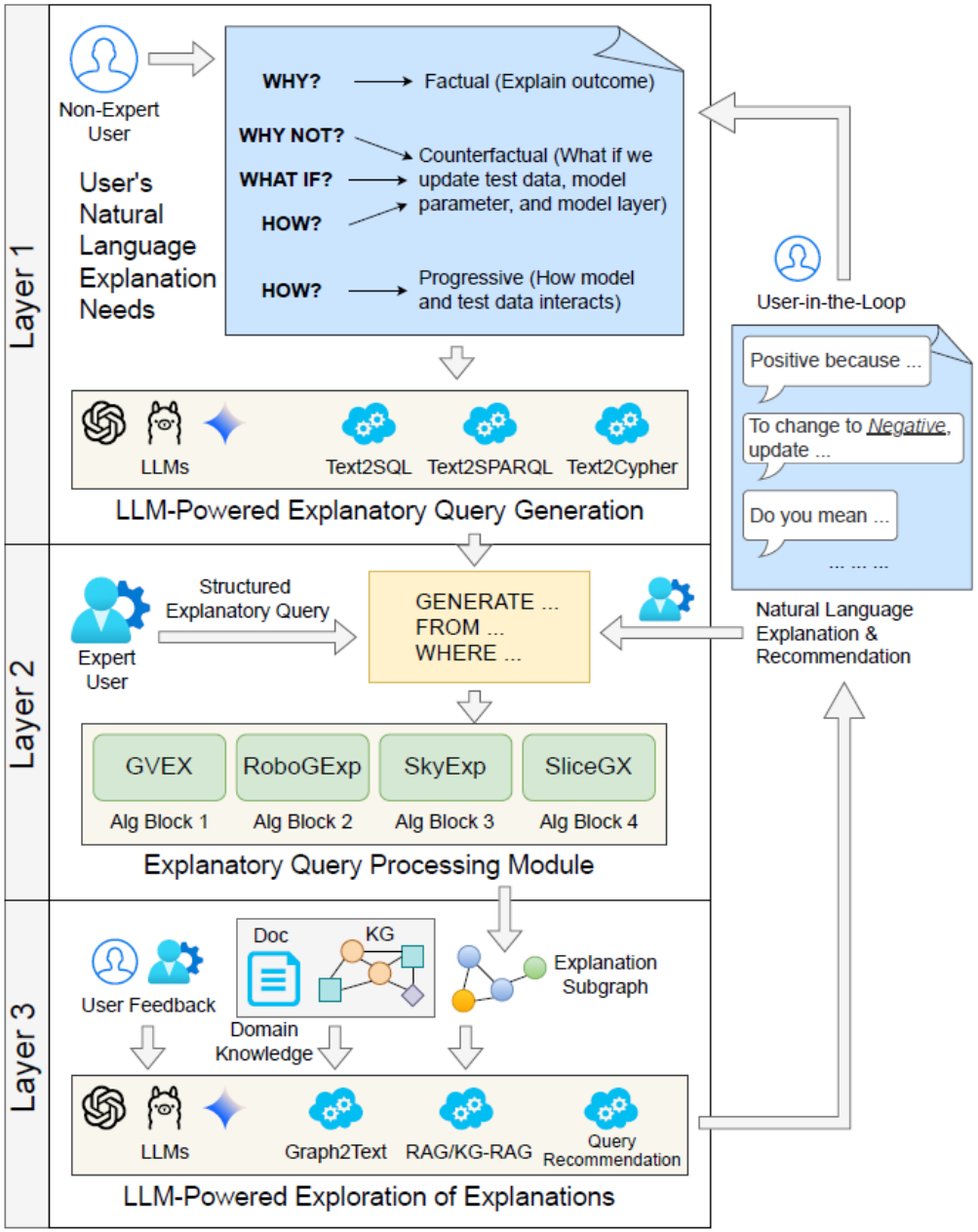
2. **Hybrid Framework (ATEX-CF):** Jointly leverages edge deletions (traditional CF) and attack-informed edge additions for more comprehensive explanations.

3. **Enhanced Explanatory Coverage:** Uncovers missing but plausible relations, complements deletion-based methods, and enables proactive graph modifications.

4. **Efficiency & Controllability:** Attack-guided candidate selection reduces combinatorial complexity; sparsity + plausibility constraints ensure realistic explanations.

5. **State-of-the-Art Results:** Wins 20/30 metric–dataset cells across 6 benchmarks, 3 GNN architectures, and 9 baselines.

GNN Explainers 2.0: User-centric and Data-driven



- Explanations for end-users and domain experts
- Layer-wise provenance for model debugging and optimization
- Interactive, configurable, efficient, and scalable explanations
- Robust and multi-criteria optimization explanations
- Explanatory query and output interface based on structured query, natural language, and examples

Tutorial outline

1 Introduction

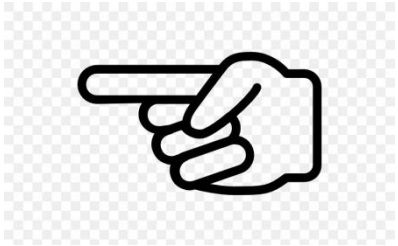
2 GNN Explainers Categorization

3 GNN Explainers 2.0

4 User-centric and Data-driven Explainability Methods for GNNs

5 Future directions

- 5.1 Downstream Tasks beyond Classification
- 5.2 Qualitative Evaluation of GNN Explanation
- 5.3 Explainability for Complex GNNs
- 5.4 Explanation with Privacy Concern
- 5.5 Explanation as Actionable Recourse
- 5.6 Multi-modal Explanation

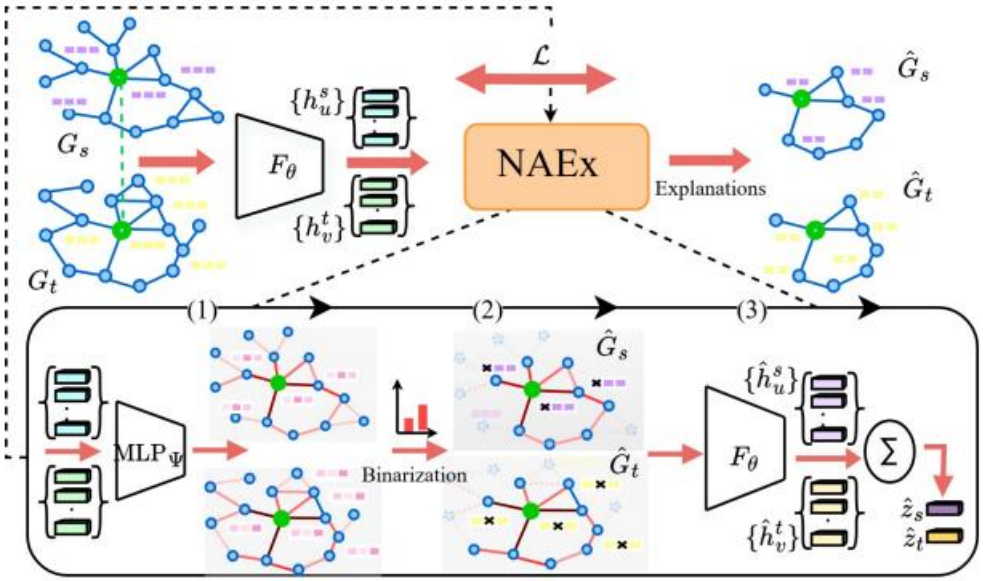


Future Directions

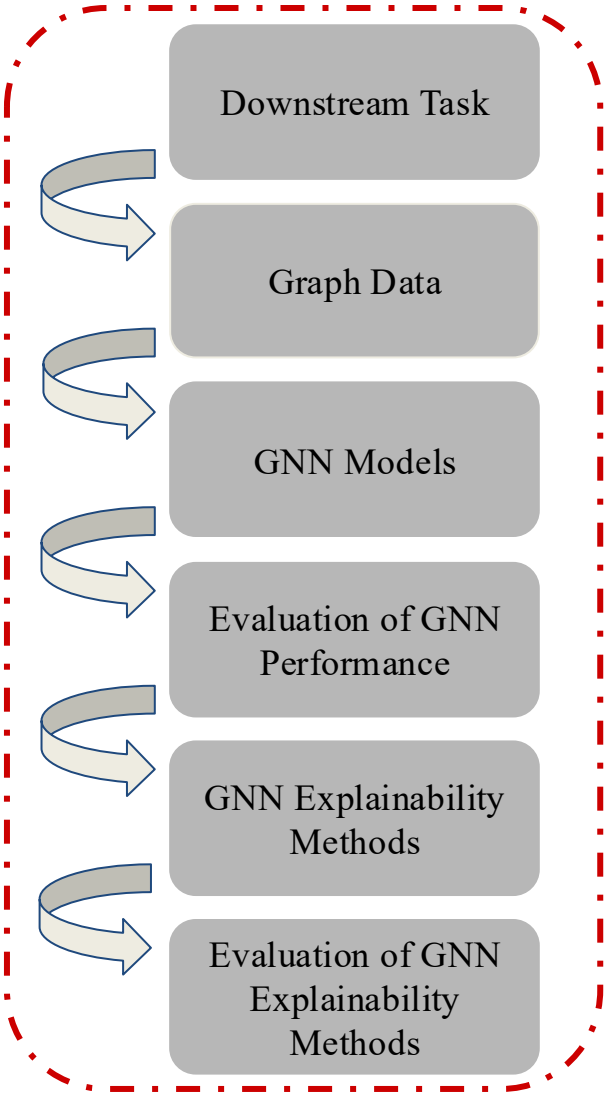
Arijit Khan

Downstream Tasks Beyond Classification

- Benchmarking interplay of GNN models, graph data, explainability methods, evaluation metrics, and downstream tasks.
- Downstream tasks beyond graphs and nodes classification.
  - Link prediction/ recommendation
  - Question answering
  - Network alignment
  - Task-agonistic GNN explanation



The NAEx framework for Network Alignment Explanation



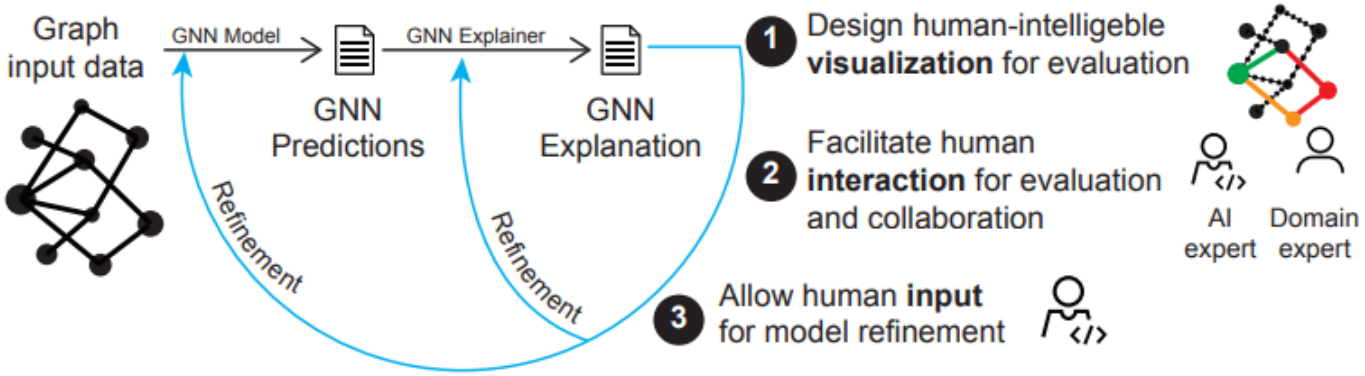
Paper: https://arxiv.org/abs/2508.04731

NAEx: A Plug-and-Play Framework for Explaining Network Alignment.
Shruti Saxena, Arijit Khan, Joydeep Chandra

Qualitative Evaluation of GNN Explanation

• Qualitative evaluation of GNN interpretation – usability, interactive-ness, querying with domain knowledge, trustworthiness, deployment, visualization and HCI tools.

• Obtain real-world ground truth.



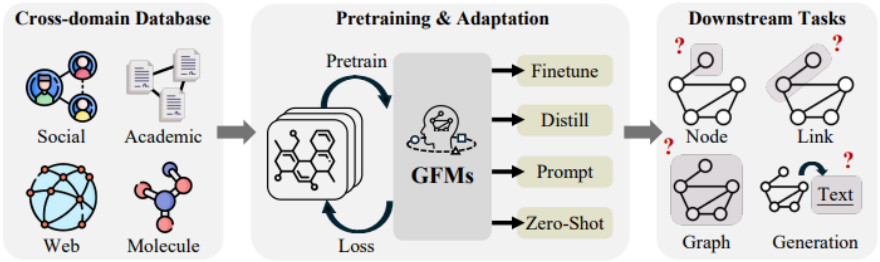
A human-centered approach toward offering GNN explanations must include human-intelligible visualizations, a user interface facilitating human interactions with visualized explanations, and finally allow human input during or immediately following such evaluation to refine either the GNN model, explainer, or both.

Paper: https://arxiv.org/abs/2405.06917

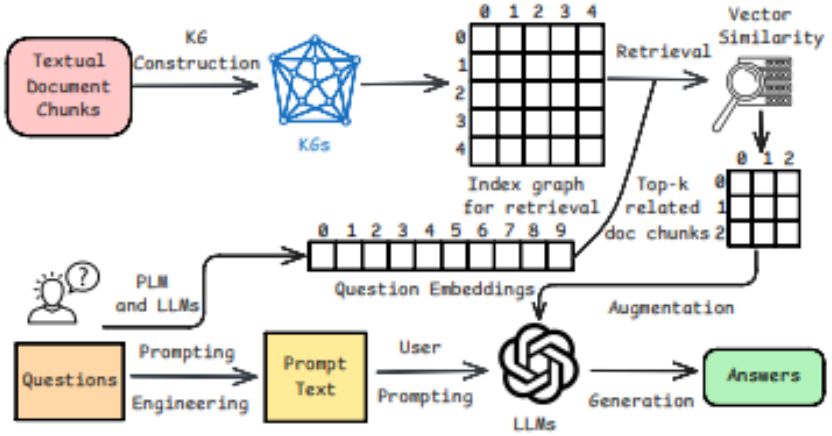
Pantea Habibi, Peyman Baghershahi, Sourav Medya, Debaleena Chattopadhyay: Design Requirements for Human-Centered Graph Neural Network Explanations.

Explainability for Complex GNNs

- Explainability for more complex graph neural networks, e.g., hypergraph neural networks, temporal graph neural networks, graph transformers, graph foundation models, and Graph RAG.



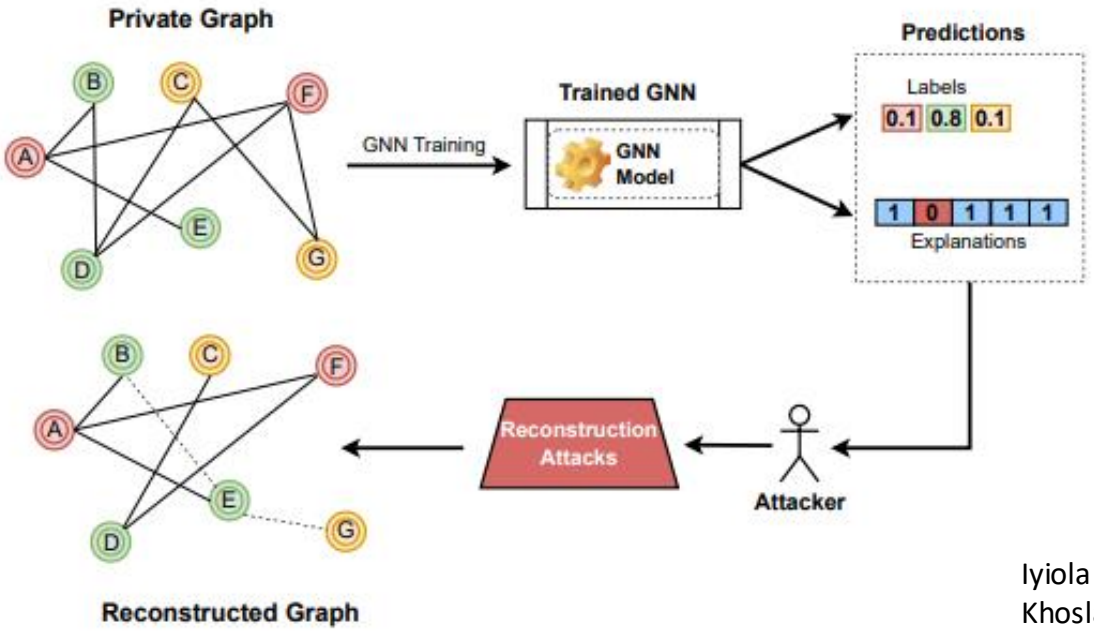
Graph Foundation Model



Graph-based Retrieval-augmented Generation (Graph RAG)

Explanation with Privacy Concern

- Privacy and explainability → two important ingredients for trustworthy ML.
- Balance explanation quality with data and model privacy.



The importance scores for the features, as provided by an explanation, can be exploited to infer the graph structure

Iyiola E. Olatunji, Mandeep Rathee, Thorben Funke, Megha Khosla. Private Graph Extraction via Feature Explanations. Proc. Priv. Enhancing Technol. 2023 https://petsymposium.org/popets/2023/popets-2023-0041.pdf

Megha Khosla. Privacy and Transparency in Graph Machine Learning: A Unified Perspective. https://arxiv.org/pdf/2207.10896

Explanation as Actionable Recourse

- Counterfactual explanation to offer actionable recourse.

- Alignment with specific domain constraints. Contextually relevant, practical, and interpretable explanations.

- In drug discovery, an identified molecular fragment might be nonexistent or unstable under laboratory conditions.

Dataset	RCExplainer			CF^2		
	Expected Count	Observed Count	p-value	Expected Count	Observed Count	p-value
Mutagenicity	233.05	70	< 0.00001	206.65	0	< 0.00001
Mutag	11	9	0.55	4	1	0.13
AIDS	17.6	8	< 0.00001	1.76	0	0.0001

Statistical significance of deviations in the number of connected graphs between the test set and their corresponding counterfactual explanations on molecular datasets. Statistically significant deviations with p-value< 0.05 are highlighted.

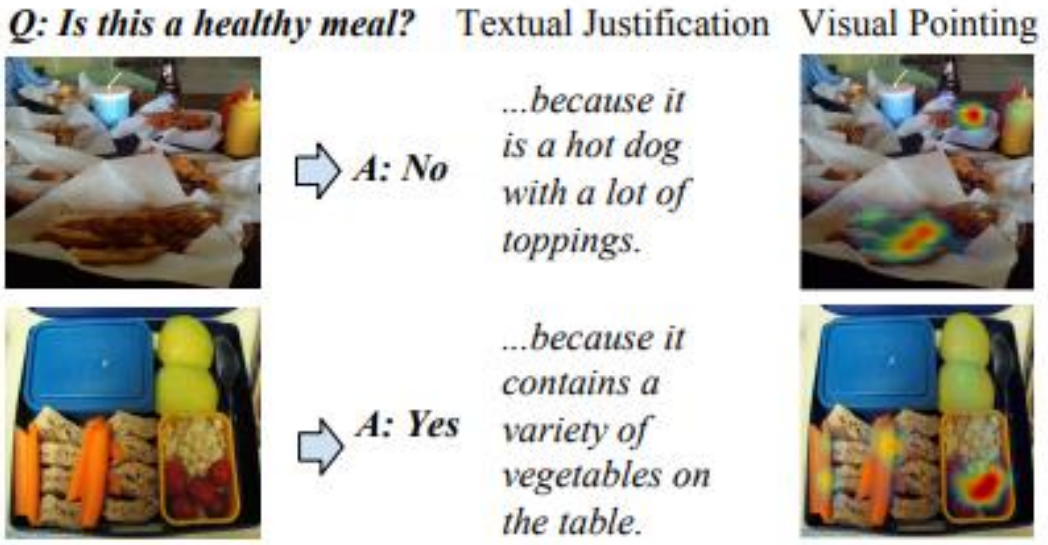
GNNX-BENCH: Unravelling the Utility of Perturbation-based GNN Explainers through In-depth Benchmarking
Mert Kosan, Samidha Verma, Burouj Armgaan, Khushbu Pahwa, Ambuj Singh, Sourav Medya, Sayan Ranu. ICLR 2024
https://arxiv.org/pdf/2310.01794

Multi-modal Explanations

• In healthcare and spatio-temporal analysis, data are multi-modal—comprising text, images, tables, and multimedia.

• Graph-based models, like knowledge graphs, offer integrated solutions for multi-modal data.

• GNN-LLM collaborative systems can process and generate outputs across these modalities.



Multimodal Explanations: Justifying Decisions and Pointing to the Evidence. Dong Huk Park , Lisa Anne Hendricks, Zeynep Akata , Anna Rohrbach, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach https://arxiv.org/abs/1802.08129

Thanks!

Website - https://gnn-explainers.github.io/

arijitk@bgsu.edu; xiangyu.ke@zju.edu.cn; yxw1650@case.edu; bonchi@gmail.com

Thanks for Website Setup and Logo Design!
Mengying Wang mxw767@case.edu