Advances and Future Perspectives in Graph Counterfactual Explanations

EDBT 2026

Prof. Giovanni Stilo
Luiss Guido Carli University – Rome, Italy
gstilo@luiss.it

LUISS

EDBT ICDT 2026 TAMPERE

aiimlab.org

LUISS GUIDO CARLI UNIVERSITY AND BUSINESS SCHOOL

Founded in 1974 in Rome, LUISS is a leading private university specializing in social sciences, supported by Confindustria — Italy's main confederation of industries.

With five campuses in the heart of Rome, LUISS offers world-class education in Law, Business & Management, Economics & Finance, and Political Science, and Artificial Intelligence and Decision Science, combining rigorous academics with deep connections to the business world.

STUDENTS
12,000+
Multicultural community from 117 countries

DEPARTMENTS
5
Law, Business, Economics & Political Science, AI

PARTNERS
350+
University partners worldwide

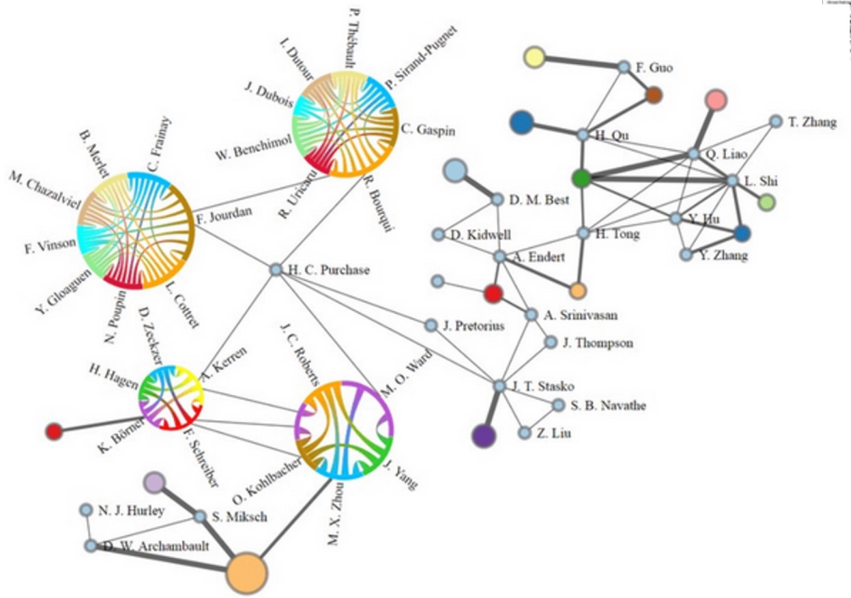
EMPLOYMENT
90%
Graduate employment within first year

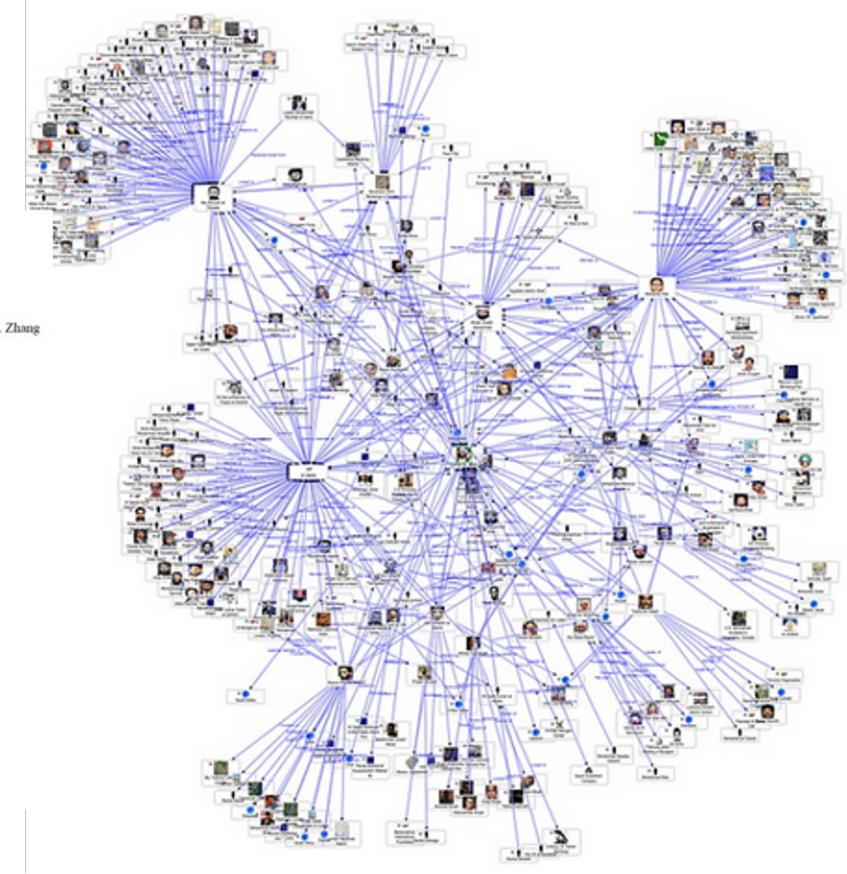
EDBT ICDT 2026 TAMPERE

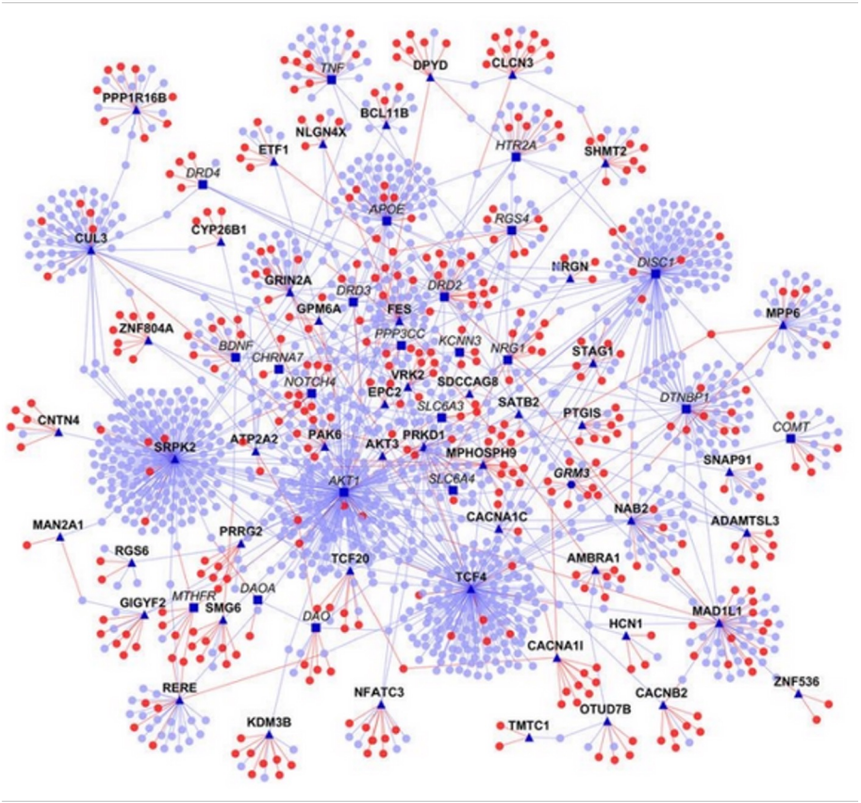
Graphs fundamentals and their Neural Network

images based on Understanding Deep Learning - book by Simon J.D. Prince

What is a graph?







Co-Authorship Networks

Social Networks

Proteins interactions

Challenges with graphs

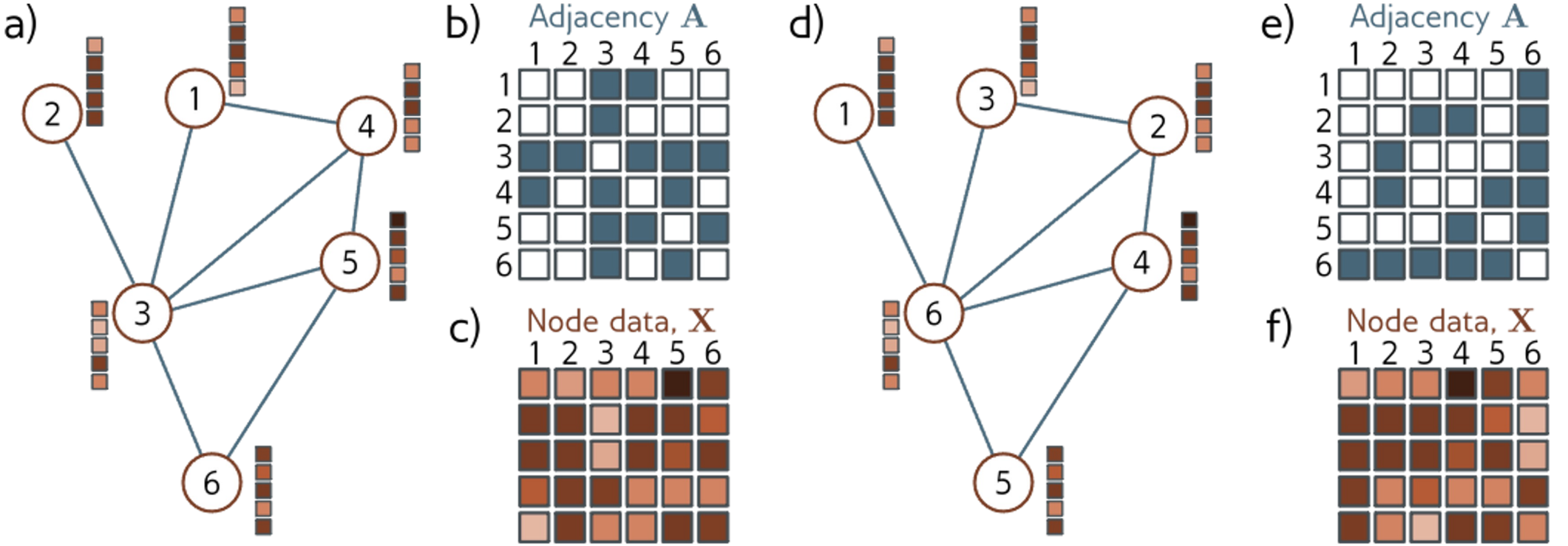
There are **three main challenges** associated with processing graphs:

- **Variable topology:** hard to design an NN that is sufficiently expressive and can cope with this variation
- **Huge graphs:** we can have millions of nodes and billions of edges (see Twitter)
- **Single monolithic graph:** the usual protocol of training with many data examples and testing with new data is not always appropriate or possible

Nodes Permutation

Node indexing in graphs is arbitrary

Permuting the node **indices** results in a **permutation** of the **columns** of the node data matrix $\boldsymbol{X}$ and a permutation of both the **rows** and **columns** of the adjacency matrix $\boldsymbol{A}$.



$$\mathbf{X'=XP}, \quad \mathbf{A'=P^T AP}$$

Graph Learning

We want to learn a (dense) representation ***H*** of the graph usable for different downstream tasks

A **graph neural network** is a model that takes:

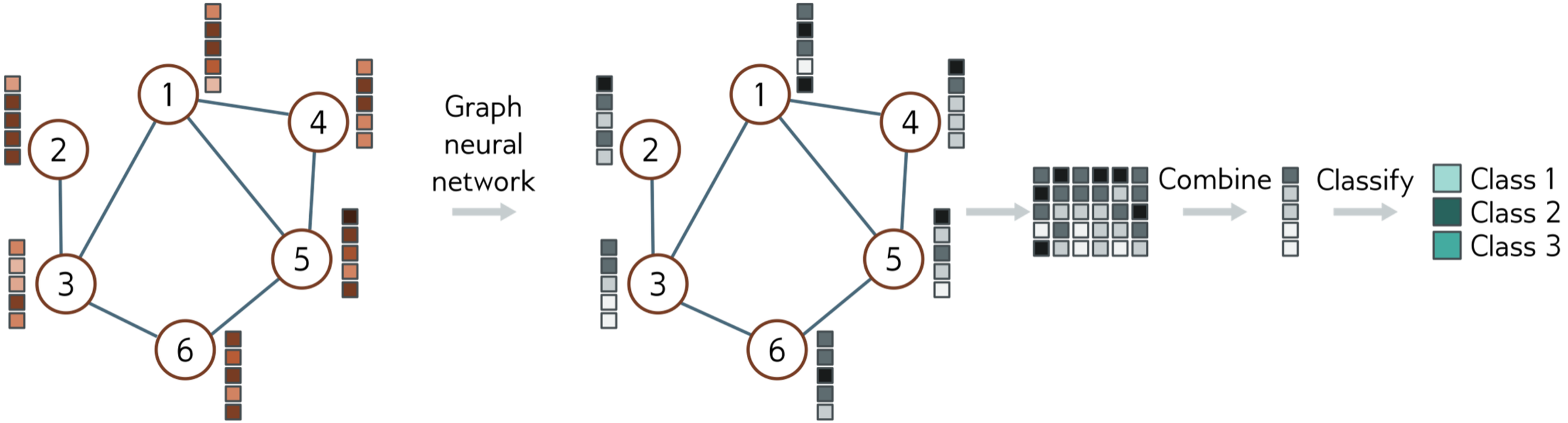
- the node embeddings ***X*** and the adjacency matrix ***A*** as inputs and passes them through a series of *k* layers.
- the node embeddings are updated in each layer to create intermediate "hidden" representations ***h*** before finally computing output embeddings h_K.

Graph Classification Tasks

For example, if we want to **predict:**

- the **temperature** at which a **molecule** becomes **liquid** (a **regression task)**;
- whether a **molecule** is **poisonous** to human beings or not (a **classification task**).

For graph-level tasks, *the output* ***node embeddings are combined*** *(e.g., by averaging), and the resulting vector is* ***mapped*** *via a linear transformation or neural network to a* ***fixed-size vector****.*

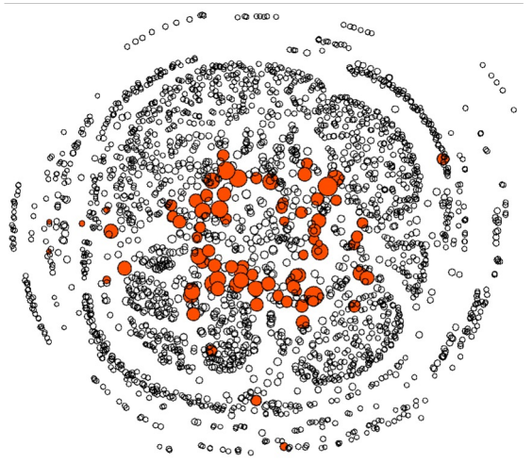


$$Pr(y = 1 \mid \boldsymbol{X}, \boldsymbol{A}) = sig\left(\beta_k + \boldsymbol{\omega}_k \boldsymbol{H}_k \frac{\mathbf{1}}{N}\right)$$

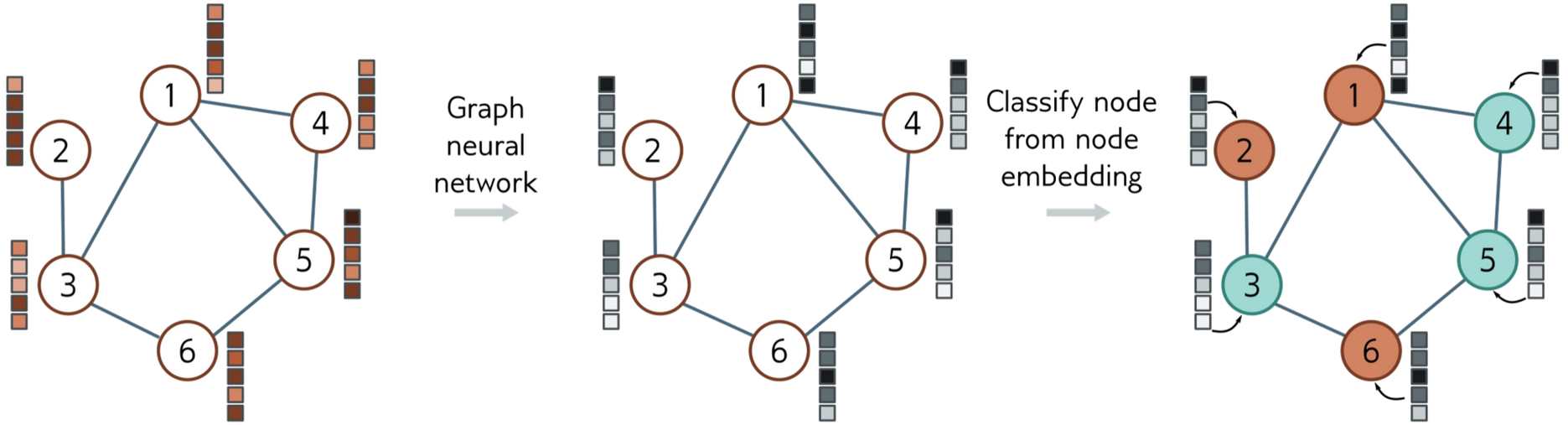
Node Classification Tasks

For example, in an *PPI* network we might want to **predict:**

- the probability that a **given node** might be **attacked/being part** of a certain **disease** (classification) as it is shown for COVID19 (**red**) - PPI (on the right).



*The network assigns **one or more label** (classification) or values (regression) to **each node** of the graph, **using** both the **graph structure** and learned **node embeddings**.*

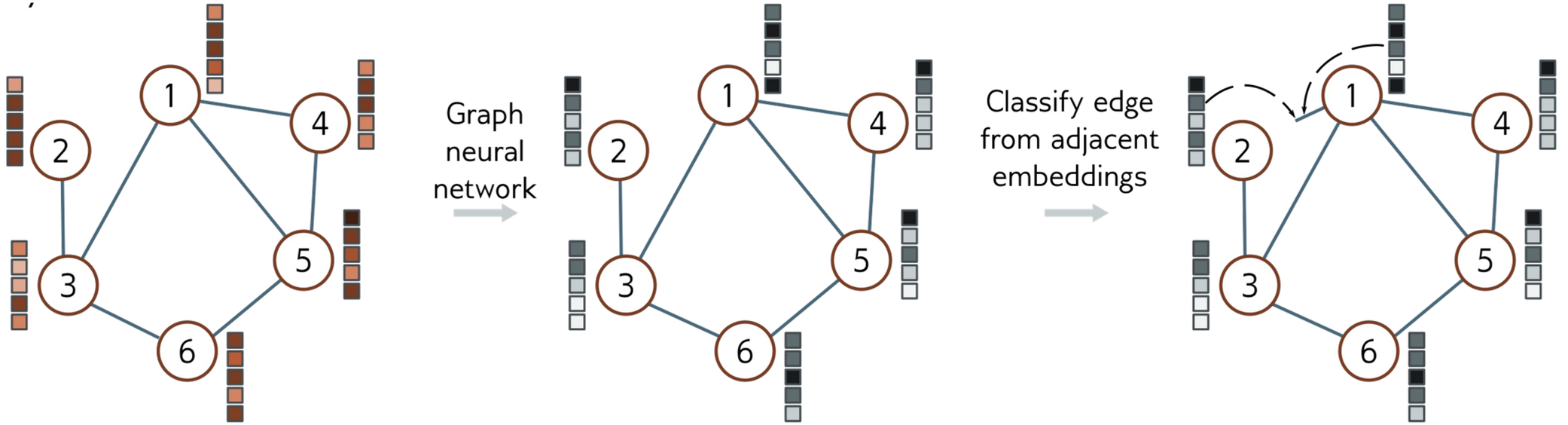


$$Pr\left(y^{(n)} = 1 \mid \boldsymbol{X}, \boldsymbol{A}\right) = sig\left(\beta_k^{(n)} + \boldsymbol{\omega}_k^{(n)} \boldsymbol{H}_k^{(n)}\right)$$

Edge Classification Tasks

For example, in the **social network** setting, we want to **predict** whether two people know each other and suggest that they connect if that is the case (recommendation).

*The network assigns **one or more label** (classification) or values (regression) to **each edges** of the graph, using both the graph **structure** and learned **node embeddings**.*



$$Pr\left(\boldsymbol{y}^{(mn)} = 1 \mid \boldsymbol{X}, \boldsymbol{A}\right) = sig\left(\boldsymbol{H}_k^{(m)T} \cdot \boldsymbol{H}_k^{(n)}\right)$$

GCN (1)

each node at layer *k*, we aggregate information from neighboring nodes by e.g. summing their node embeddings:

$$\boldsymbol{agg}[n,k] = \sum_{m \in \mathrm{ne}[\mathrm{n}]} \boldsymbol{H}_k^{(m)}$$

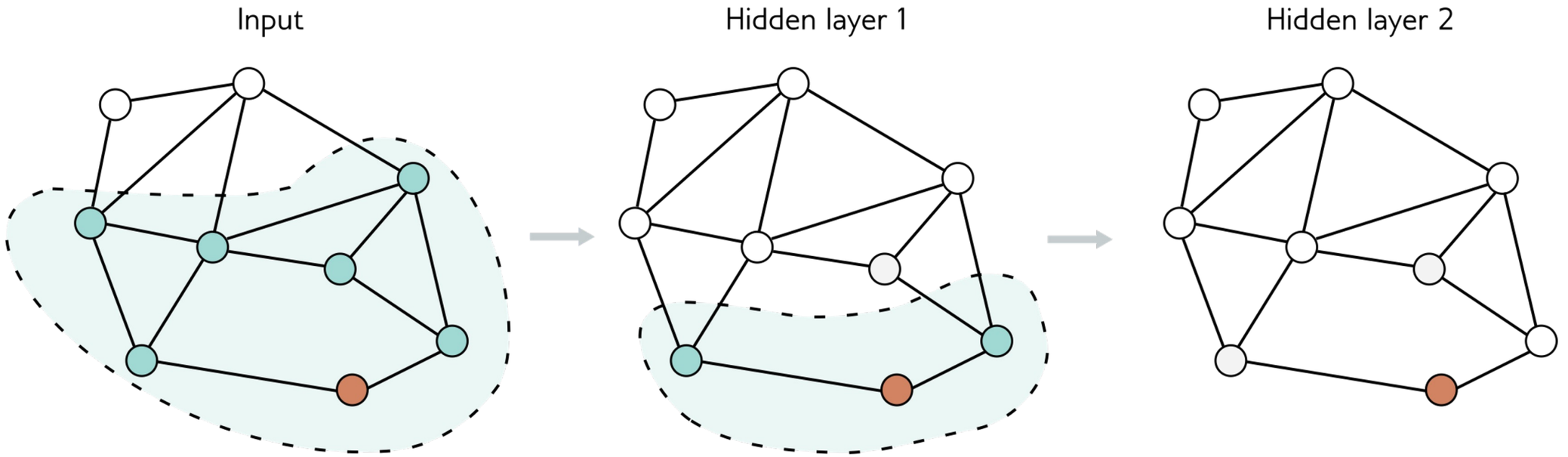
linear transformation **Ω** to the embedding H of the current node and to his aggregated value, we add a bias term β, and pass the result through a nonlinear activation function $a[\cdot]$, which is applied independently to every member of its vector argument:

$$\boldsymbol{H}_{k+1}^{(n)} = a\left[\boldsymbol{\beta}_k + \boldsymbol{\Omega}_k \cdot \boldsymbol{H}_k^{(n)} + \boldsymbol{\Omega}_k \cdot \boldsymbol{agg}[n,k]\right]$$

the n^{th} column of ***A*** contains ones at the positions of neighbors. If **post-multiply** the **embeddings** by **A** the n^{th} column is **agg**[n,k]:

$$\boldsymbol{H}_{k+1} = a[\boldsymbol{\beta}_k \mathbf{1}^T + \boldsymbol{\Omega}_k \boldsymbol{H}_k + \boldsymbol{\Omega}_k \boldsymbol{H}_k \boldsymbol{A}] = a[\boldsymbol{\beta}_k \mathbf{1}^T + \boldsymbol{\Omega}_k \boldsymbol{H}_k (\boldsymbol{A} + \boldsymbol{I})]$$

GNNs (BRIEFLY)



$$H^{\ell} \text{ depends on } X(A+I)^{\ell}$$

graph expansion problem

If there are **many layers** and the graph is **densely connected:**

every input node may be in the receptive field of every output.

In general we want that k << diam(G)

eXplainable Artificial Intelligence

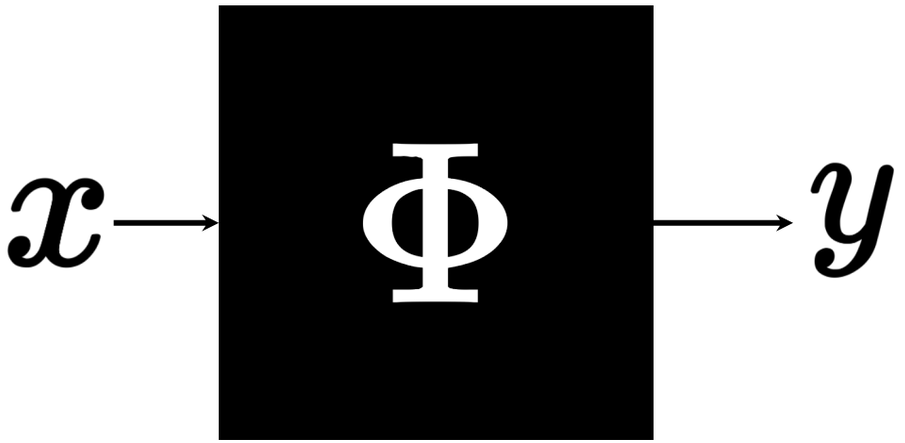
first part of the slides based on: CSEP 590B: Explainable AI from University of Washington

What's going on today in ML?

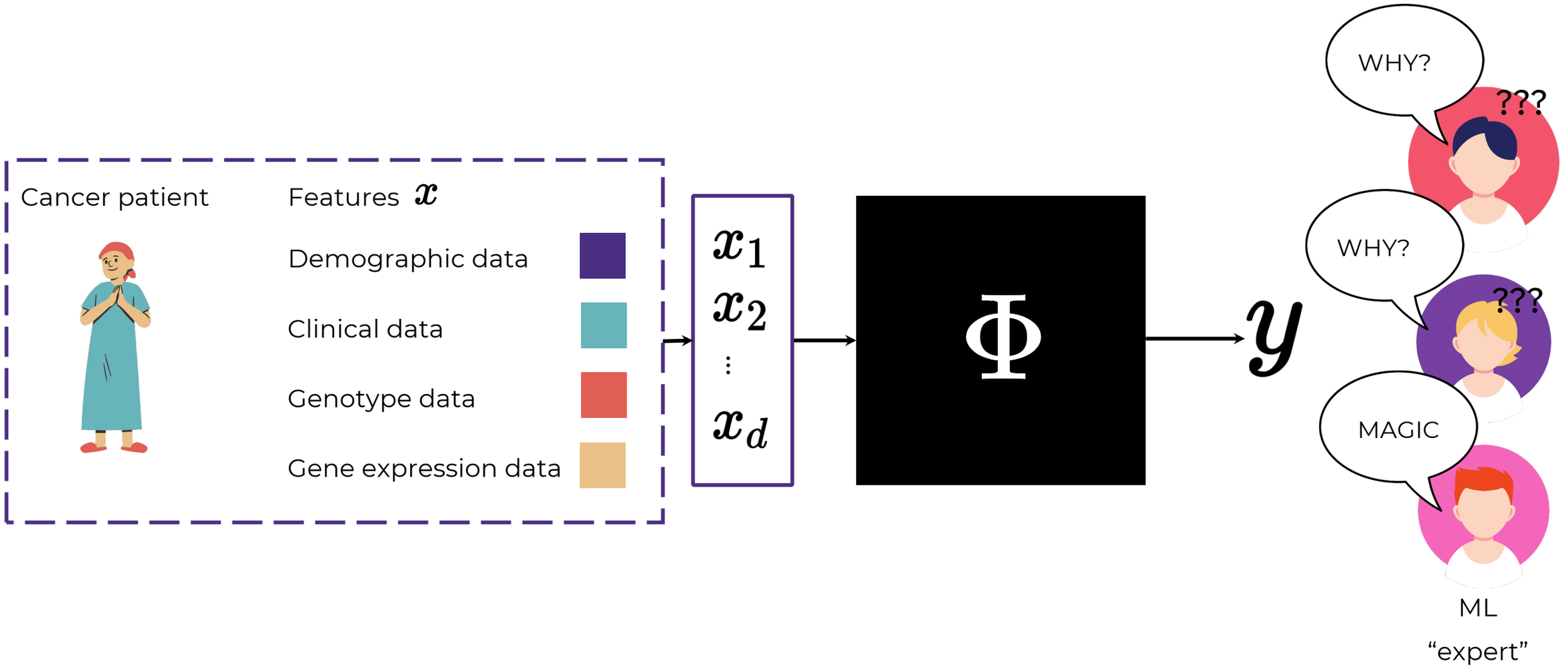
Lack of transparency

- Identify key factors in underlying processes

- Generate scientific hypotheses



WHY ACCURATE PREDICTIONS ARE IMPORTANT?



Explainability

- Which features contributed to a certain prediction and how?
- How to learn or select features that are most interpretable or informative?
- How to make biological or clinical sense of a black-box model?

Example: Credit Risk Evaluation

- We work at a bank, and our boss is asking us to automate **credit risk evaluation**
- **Step 1**: we get some historical data

$$x = [\text{age}, \text{job}, \text{salary}, \text{sex}, \ldots]$$

$$y \in \{\text{approved}, \text{declined}\}$$

- **Step 2**: train a model, complex one because it gets the best performances out of 100 different models we trained and evaluated

Question #1

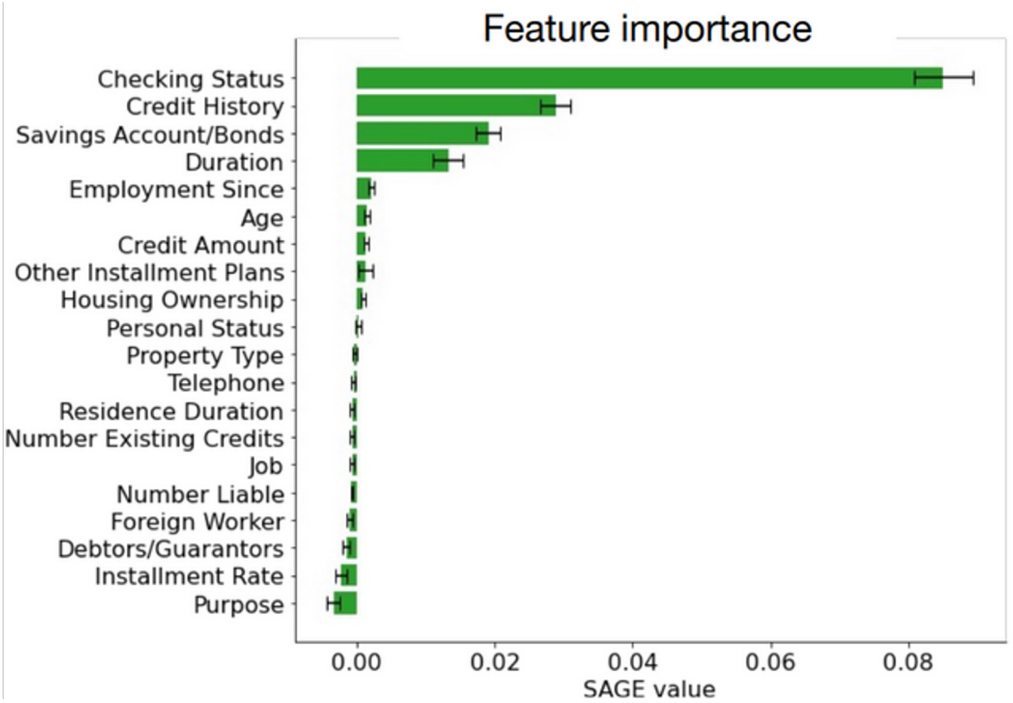
What did the model learn, and how does it make decisions?

- We can’t easily summarize the patterns, rules, concepts learned by a complex model
- Model have too many parameters to examine
- Coarse summary: count number of splits on each feature

Question #2 – Global Importance

Which features are most important overall?

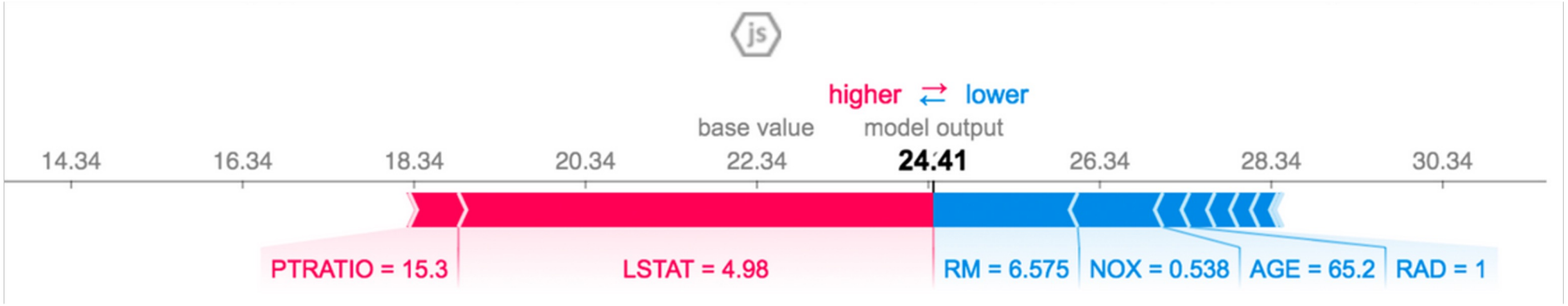
- Can be answered by permutation tests (Breiman, 2001)
- There are in fact many methods for analyzing **global feature importance**



Question #3 – Factual Explanation

For the customers whose loans are denied, can we tell which features led to the decision?

- In this case, must analyze individual predictions (not overall behavior)
- Many methods designed to assess **local feature importance**



Types of Explainability

Feature importance explanations

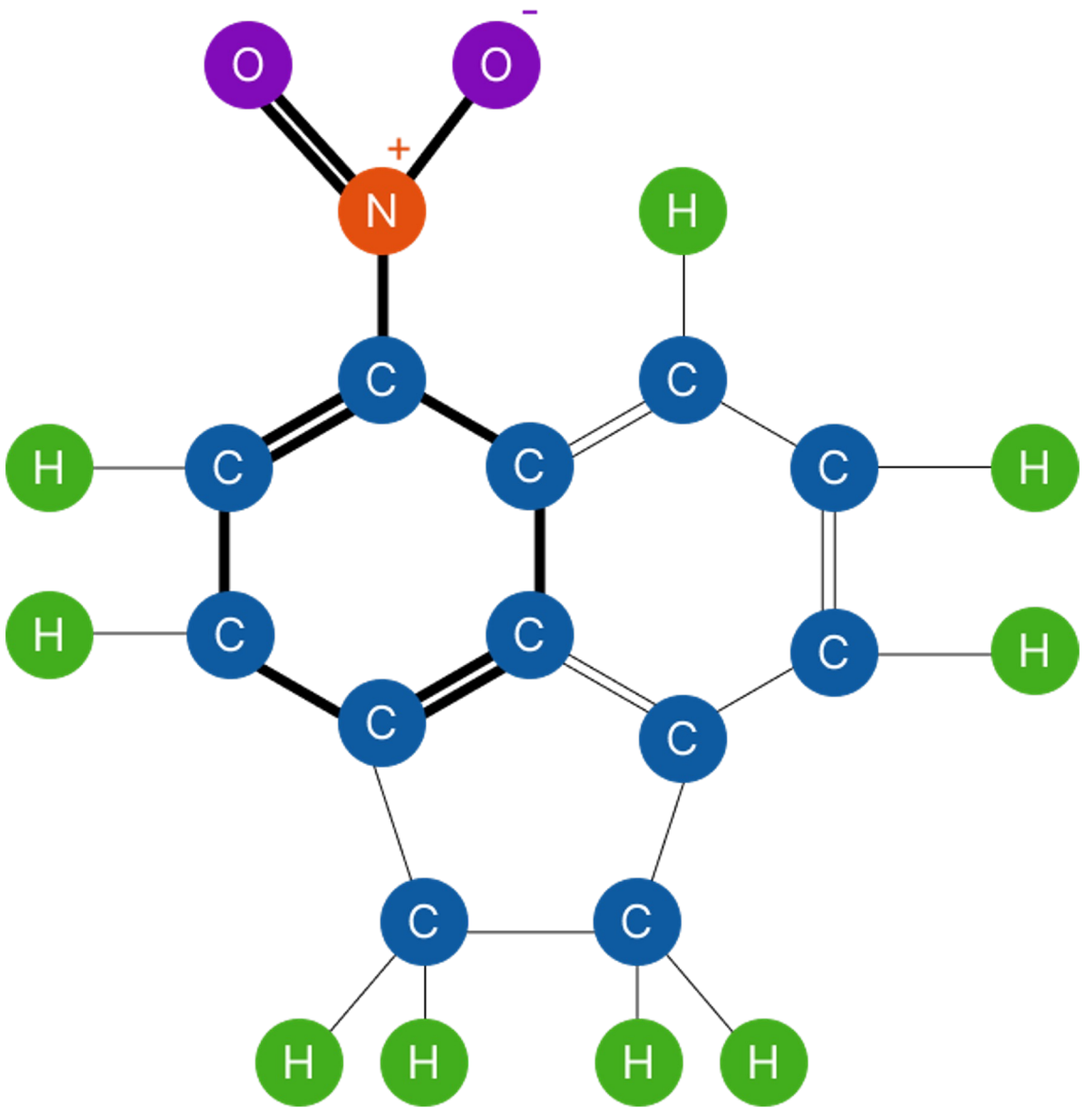
- Removal-based explanations
- Shapley values
- Propagation-based explanations

Some sort of factual explanations

Inherently interpretable models

Counterfactual explanations

Factual explanations on Graphs



Factual explanations on Graphs

- Find the subgraph has the same label as the whole graph (**desideratum #1**)
- This subgraph should be minimal (**desideratum #2**)
- When you remove this subgraph, the remainder should have the opposite class (**desideratum #3**) – *this gives sprout to factual-based counterfactual explainers*

COUNTERFACTUAL EXPLANATIONS IN GRAPHS

Based on:

Prado-Romero et al. “A survey on graph counterfactual explanations: definitions, methods, evaluation”, ACM CSUR 2024

Graph Counterfactual Explanation

Multi-class minimal counterfactual examples: Let Φ be a prediction model that classifies x into a class $c \in C$ from a set of classes C. Let X' be the set of possible counterfactual examples x' and $S_{inst}(x, x')$ be a similarity measure that tells how similar x' is to x. Then, we define the set of counterfactual examples w.r.t. Φ as follows:

$$s(c', x) := \max_{x' \in X', x \neq x'} \{\mathcal{S}_{inst}(x, x') \mid \Phi(x') = c'\}$$

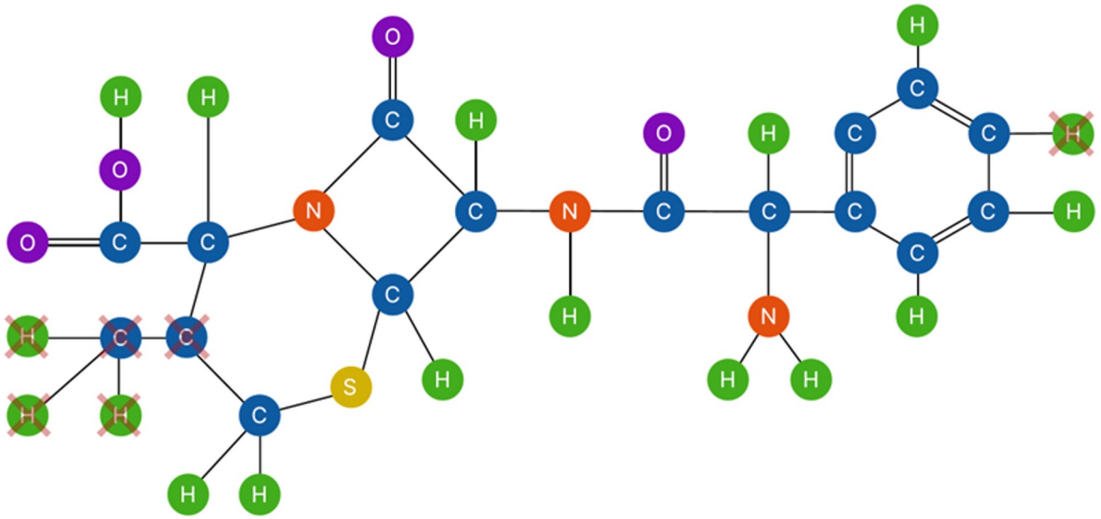
$$\mathcal{E}_\Phi(x) = \bigcup_{c' \in C - \{c\}} \{x' \in X' \mid x \neq x', \mathcal{S}_{inst}(x, x') = s(c', x)\}$$

Global minimal counterfactual example: Let Φ be a prediction model that classifies x into a class $c \in C$. Let X' be the set that contains all the possible counterfactual examples x'. We define the global minimal counterfactual example $\mathcal{E}^*_\Phi(x)$ of x, as follows:

$$\mathcal{E}^*_\Phi(x) = \arg\max_{x' \in X'} \mathcal{S}_{inst}(x, x')$$

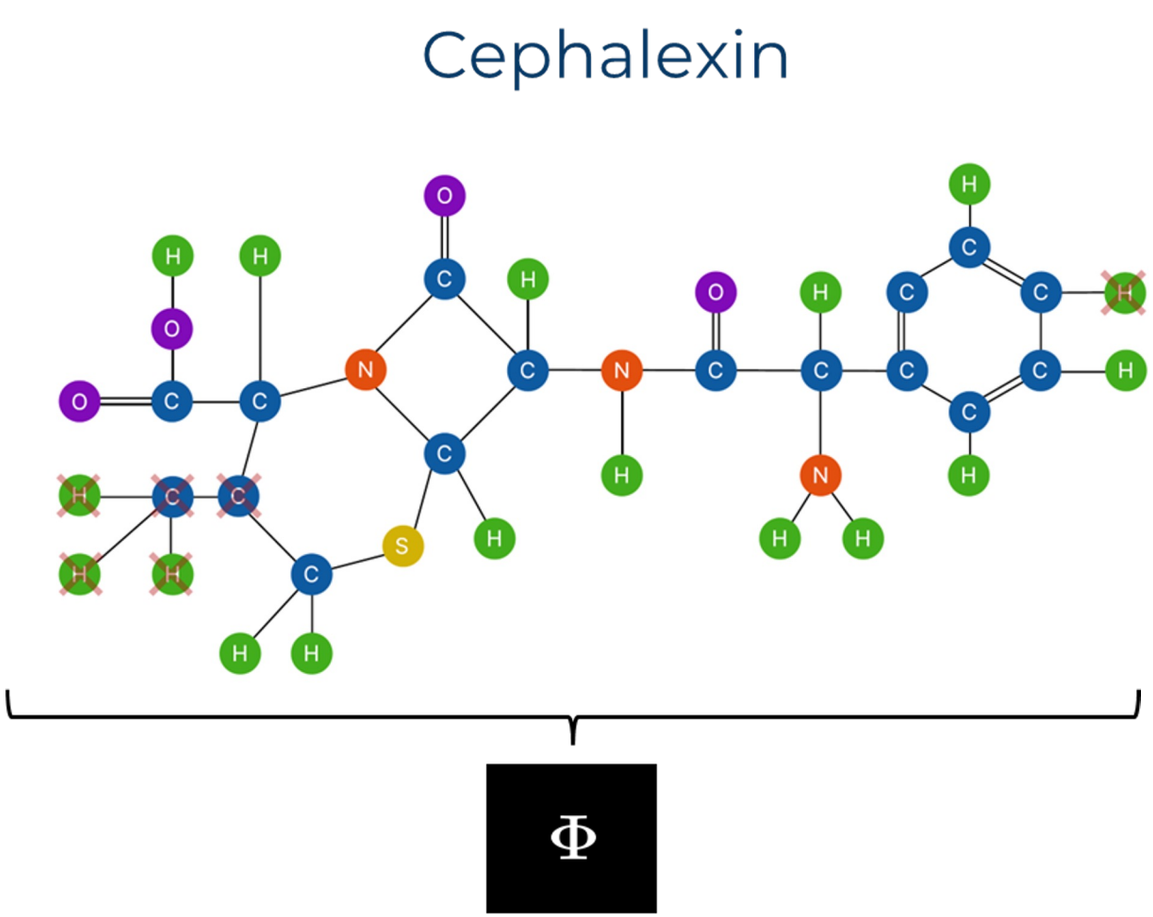
GCE for Drug Discovery

Cephalexin



Bacterial Infection

GCE for Drug Discovery

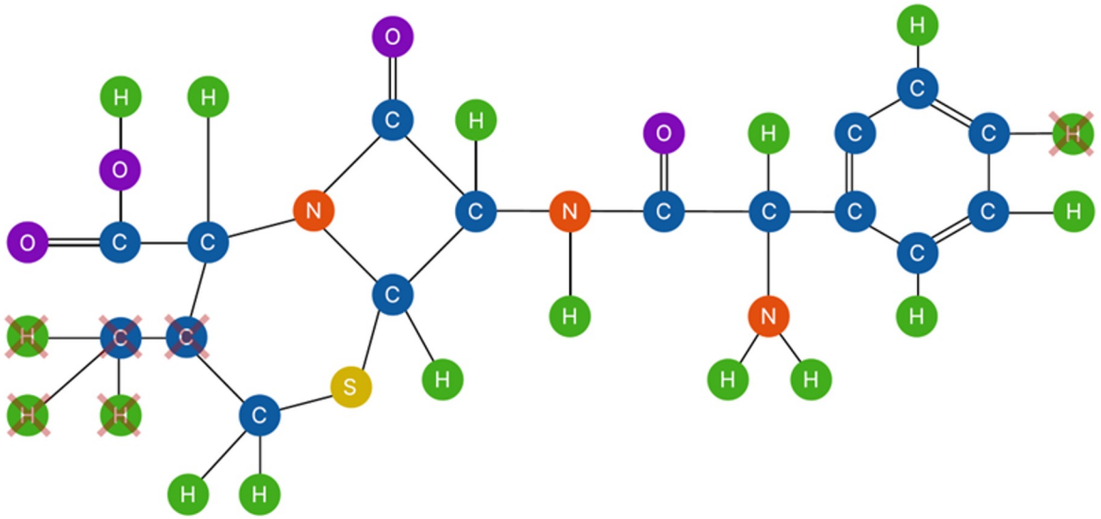


Headache

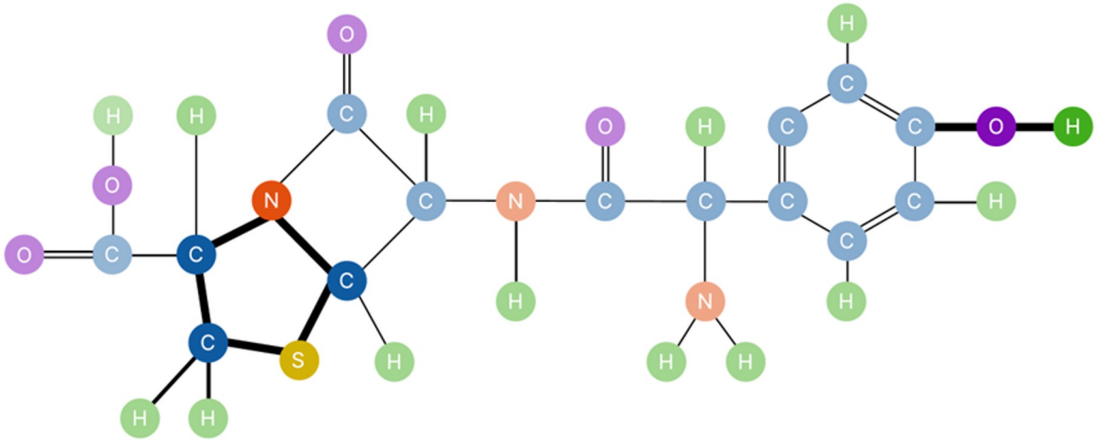
Bacterial Infection

GCE for Drug Discovery

Cephalexin



Amoxicillin

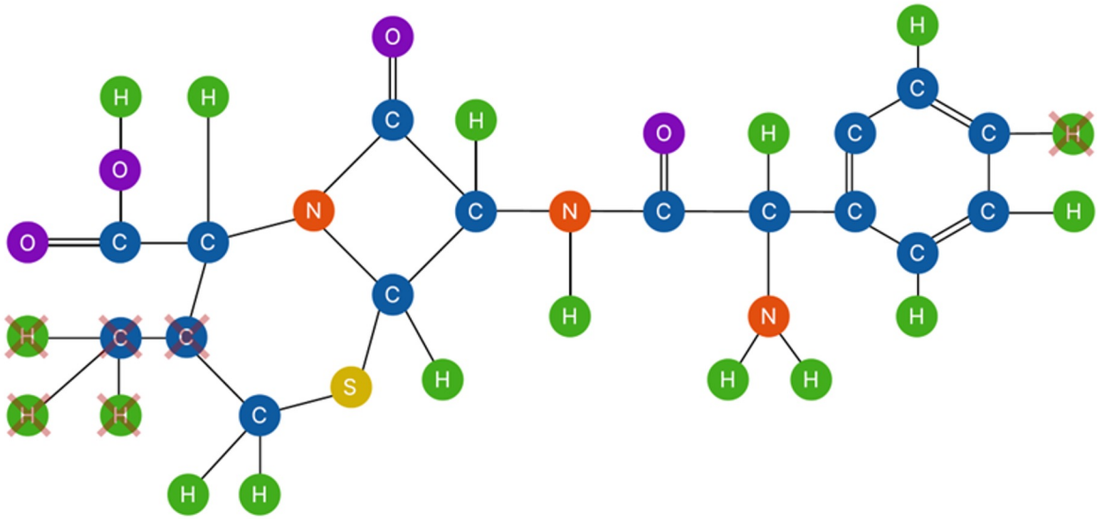


Headache

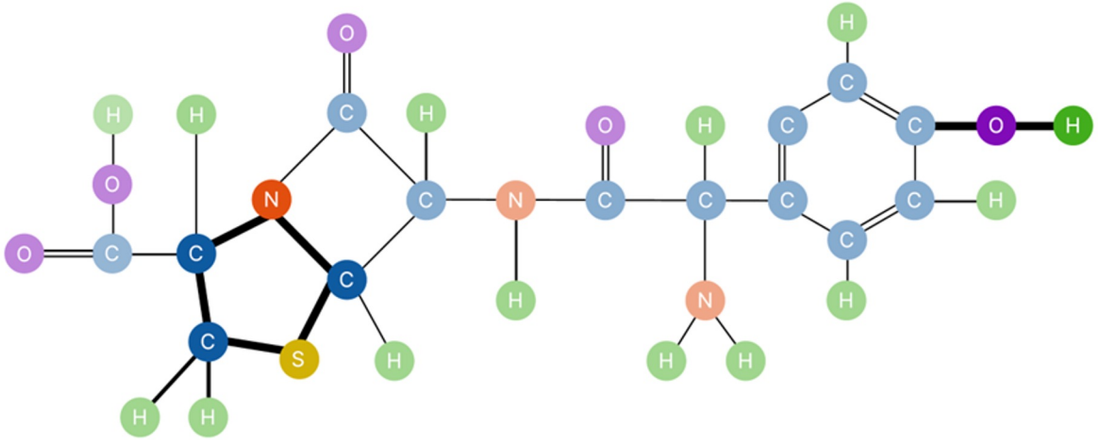
Bacterial Infection

GCE for Drug Discovery

Cephalexin



Amoxicillin



Headache

Bacterial Infection

No Headache

Challenges in generating GCEs

- Structural Constraints
- Ensuring plausibility and feasibility of the counterfactual
- Computational complexity

Evaluating Graph Counterfactual Explanations

EVALUATION DATASETS

Synthetic Datasets

- Designed with simple, interpretable structures
- **Examples:** Tree-Cycles, Tree-Grid, Tree-Infinity, BA-Shapes, BA-Community, BA-2motifs
- **Characteristics:**
  - Small-medium graphs with ground-truth motifs
  - Ideal for testing GNN explainability accuracy
  - High interpretability, low noise
- **Use Case:** Benchmarking explainers on well-understood graph patterns

Overview of GCE benchmarking datasets

Simple Synthetic data with ground truth

Dataset	Domain	Publicly Available Repository (Data or Code)	Used by
Tree-Cycles [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[70, 73, 77, 113]
Tree-Grid [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[73, 77, 113]
Tree-Infinity	synthetic	https://github.com/MarioTheOne/GRETEL	[37]
BA-Shapes [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[70, 73, 77, 113]
BA-Community [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[77]
BA-2motifs [98]	synthetic	https://github.com/flyingdoog/PGExplainer	[70, 77]
ADHD [134]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/adhd	[80]
ASD [135, 136]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/autism/asd	[80]
BBBP [148]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=bbbp.zip	[120]
HIV [137–139]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=hiv.zip	[120, 123]
Ogbg-molhiv [140]	molecular	https://huggingface.co/datasets/OGB/ogbg-molhiv	[79]
Mutagenicity [141]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/Mutagenicity.zip	[70, 77, 123]
NCI1 [143]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/NCI1.zip	[70, 77, 123]
TOX21 [142]	molecular	https://tripod.nih.gov/tox21/challenge/data.jsp	[75]
ESOL [144]	molecular	https://github.com/deepchem/deepchem	[70, 75]
Proteins [145]	molecular	https://chrsmrrs.github.io/datasets/docs/datasets/	[123]
Davis [149]	molecular	http://staff.cs.utu.fi/~aatapa/data/DrugTarget/	[76]
PDBBind [150]	molecular	http://www.pdbbind.org.cn/	[76]
CiteSeer [146]	social	https://linqs.org/datasets/	[70, 112]
IMDB-M [147]	social	https://virginia.app.box.com/s/941v9pwh83lfw5vnwfbgcertlsoivg5j	[79]
CORA [151]	social	https://relational.fit.cvut.cz/dataset/CORA	[112]
Musae-Facebook [152]	social	https://www.kaggle.com/datasets/rozemberczki/musae-facebook-pagepage-network	[112]
LastFM [153]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/lastfm	[114]
Yelp [154]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/yelp2018/	[114]

Overview of GCE benchmarking datasets

Brain networks without attributes

Dataset	Domain	Publicly Available Repository (Data or Code)	Used by
Tree-Cycles [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[70, 73, 77, 113]
Tree-Grid [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[73, 77, 113]
Tree-Infinity	synthetic	https://github.com/MarioTheOne/GRETEL	[37]
BA-Shapes [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[70, 73, 77, 113]
BA-Community [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[77]
BA-2motifs [98]	synthetic	https://github.com/flyingdoog/PGExplainer	[70, 77]
ADHD [134]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/adhd	[80]
ASD [135, 136]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/autism/asd	[80]
BBBP [148]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=bbbp.zip	[120]
HIV [137–139]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=hiv.zip	[120, 123]
Ogbg-molhiv [140]	molecular	https://huggingface.co/datasets/OGB/ogbg-molhiv	[79]
Mutagenicity [141]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/Mutagenicity.zip	[70, 77, 123]
NCI1 [143]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/NCI1.zip	[70, 77, 123]
TOX21 [142]	molecular	https://tripod.nih.gov/tox21/challenge/data.jsp	[75]
ESOL [144]	molecular	https://github.com/deepchem/deepchem	[70, 75]
Proteins [145]	molecular	https://chrsmrrs.github.io/datasets/docs/datasets/	[123]
Davis [149]	molecular	http://staff.cs.utu.fi/~aatapa/data/DrugTarget/	[76]
PDBBind [150]	molecular	http://www.pdbbind.org.cn/	[76]
CiteSeer [146]	social	https://linqs.org/datasets/	[70, 112]
IMDB-M [147]	social	https://virginia.app.box.com/s/941v9pwh83lfw5vnwfbgcertlsoivg5j	[79]
CORA [151]	social	https://relational.fit.cvut.cz/dataset/CORA	[112]
Musae-Facebook [152]	social	https://www.kaggle.com/datasets/rozemberczki/musae-facebook-pagepage-network	[112]
LastFM [153]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/lastfm	[114]
Yelp [154]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/yelp2018/	[114]

Overview of GCE benchmarking datasets

Dataset	Domain	Publicly Available Repository (Data or Code)	Used by
Tree-Cycles [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[70, 73, 77, 113]
Tree-Grid [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[73, 77, 113]
Tree-Infinity	synthetic	https://github.com/MarioTheOne/GRETEL	[37]
BA-Shapes [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[70, 73, 77, 113]
BA-Community [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[77]
BA-2motifs [98]	synthetic	https://github.com/flyingdoog/PGExplainer	[70, 77]
ADHD [134]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/adhd	[80]
ASD [135, 136]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/autism/asd	[80]
BBBP [148]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=bbbp.zip	[120]
HIV [137–139]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=hiv.zip	[120, 123]
Ogbg-molhiv [140]	molecular	https://huggingface.co/datasets/OGB/ogbg-molhiv	[79]
Mutagenicity [141]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/Mutagenicity.zip	[70, 77, 123]
NCI1 [143]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/NCI1.zip	[70, 77, 123]
TOX21 [142]	molecular	https://tripod.nih.gov/tox21/challenge/data.jsp	[75]
ESOL [144]	molecular	https://github.com/deepchem/deepchem	[70, 75]
Proteins [145]	molecular	https://chrsmrrs.github.io/datasets/docs/datasets/	[123]
Davis [149]	molecular	http://staff.cs.utu.fi/~aatapa/data/DrugTarget/	[76]
PDBBind [150]	molecular	http://www.pdbbind.org.cn/	[76]
CiteSeer [146]	social	https://linqs.org/datasets/	[70, 112]
IMDB-M [147]	social	https://virginia.app.box.com/s/941v9pwh83lfw5vnwfbgcertlsoivg5j	[79]
CORA [151]	social	https://relational.fit.cvut.cz/dataset/CORA	[112]
Musae-Facebook [152]	social	https://www.kaggle.com/datasets/rozemberczki/musae-facebook-pagepage-network	[112]
LastFM [153]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/lastfm	[114]
Yelp [154]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/yelp2018/	[114]

Attributed molecular datasets without ground-truth explanations

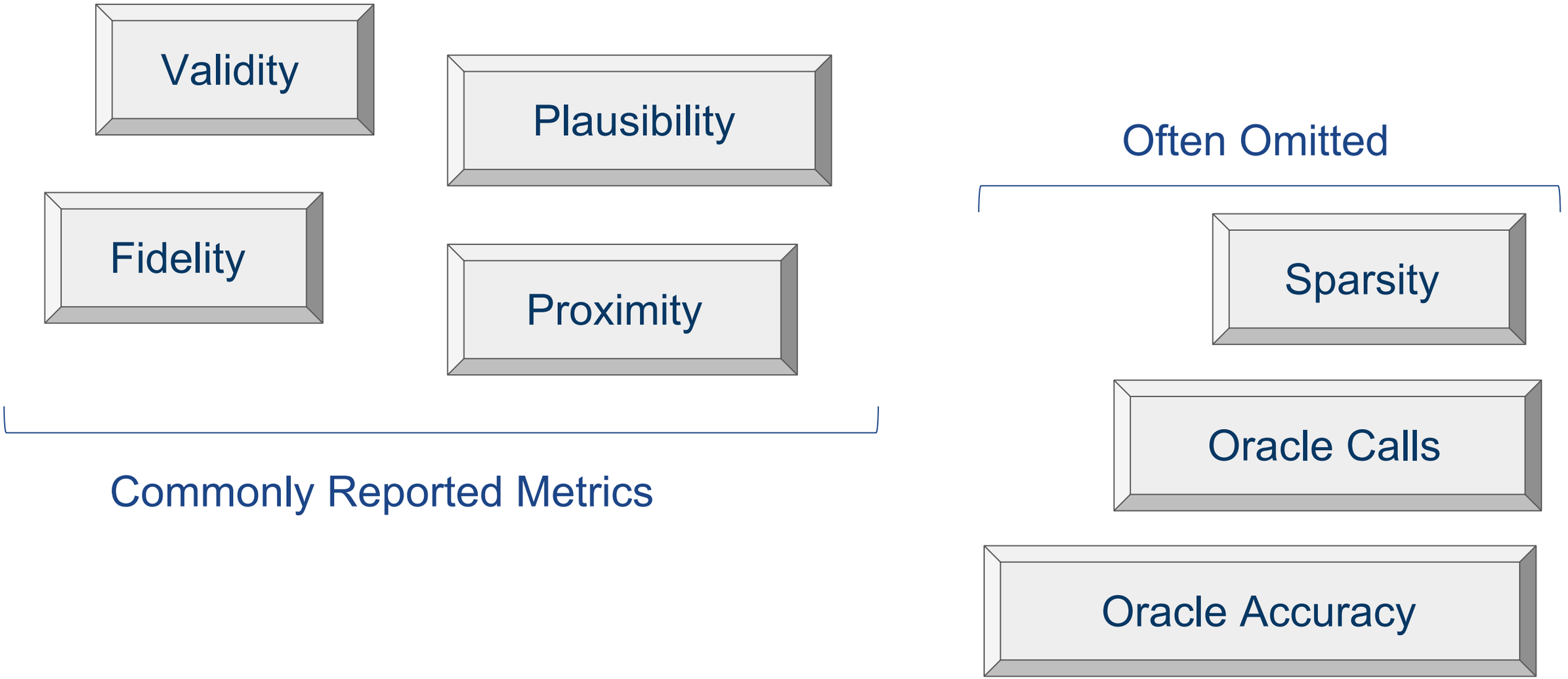
Overview of GCE benchmarking datasets

Dataset	Domain	Publicly Available Repository (Data or Code)	Used by
Tree-Cycles [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[70, 73, 77, 113]
Tree-Grid [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[73, 77, 113]
Tree-Infinity	synthetic	https://github.com/MarioTheOne/GRETEL	[37]
BA-Shapes [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[70, 73, 77, 113]
BA-Community [31]	synthetic	https://github.com/RexYing/gnn-model-explainer	[77]
BA-2motifs [98]	synthetic	https://github.com/flyingdoog/PGExplainer	[70, 77]
ADHD [134]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/adhd	[80]
ASD [135, 136]	-omics	https://github.com/MarioTheOne/GRETEL/tree/main/data/datasets/autism/asd	[80]
BBBP [148]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=bbbp.zip	[120]
HIV [137–139]	molecular	https://www.kaggle.com/datasets/mmelahi/cheminformatics?select=hiv.zip	[120, 123]
Ogbg-molhiv [140]	molecular	https://huggingface.co/datasets/OGB/ogbg-molhiv	[79]
Mutagenicity [141]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/Mutagenicity.zip	[70, 77, 123]
NCI1 [143]	molecular	https://ls11-www.cs.tu-dortmund.de/people/morris/graphkerneldatasets/NCI1.zip	[70, 77, 123]
TOX21 [142]	molecular	https://tripod.nih.gov/tox21/challenge/data.jsp	[75]
ESOL [144]	molecular	https://github.com/deepchem/deepchem	[70, 75]
Proteins [145]	molecular	https://chrsmrrs.github.io/datasets/docs/datasets/	[123]
Davis [149]	molecular	http://staff.cs.utu.fi/~aatapa/data/DrugTarget/	[76]
PDBBind [150]	molecular	http://www.pdbbind.org.cn/	[76]
CiteSeer [146]	social	https://linqs.org/datasets/	[70, 112]
IMDB-M [147]	social	https://virginia.app.box.com/s/941v9pwh83lfw5vnwfbgcertlsoivg5j	[79]
CORA [151]	social	https://relational.fit.cvut.cz/dataset/CORA	[112]
Musae-Facebook [152]	social	https://www.kaggle.com/datasets/rozemberczki/musae-facebook-pagepage-network	[112]
LastFM [153]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/lastfm	[114]
Yelp [154]	social	https://github.com/gusye1234/LightGCN-PyTorch/tree/master/data/yelp2018/	[114]

Social Networks without ground truth. Usually have many binary attributes or none

EVALUATION METRICS

Overview of evaluation Metrics



Validity

- **Definition**: Validity **checks whether a counterfactual actually flips the model's decision** relative to the original instance.
- **Formally**, for the original instance G, the counterfactual G', and oracle Φ, validity is an indicator function:

$$\Omega(G, G') = \mathbb{I}[\Phi(G) \neq \Phi(G')]$$

- **Binary signal only**: validity ignores *how* confident Φ is and the *distance* between G and G′.
- **Model access**: requires querying the oracle Φ
- **Doesn't guarantee realism**: a valid G′ may be implausible or infeasible in the data domain.

Plausibility

- We need to ensure G’ obeys **domain constraints** (e.g., valence in molecules, degree/type rules, connectivity).
- Two ways to ensure plausibility:
  - Enforce constraints during search
  - Post-hoc checks
- **Common pitfall**: unconstrained search yields infeasible yet “valid” (label-flipping) counterfactuals.

Proximity

- **Graph Edit Distance (GED):** quantifies the structural distance between the original graph G and its counterfactual G'. The distance is evaluated based on a set of actions {p1, p2, . . . , pn}, that represent a path to transform G into G'. Each action p_i is associated with a $\omega(p_i)$ cost

$$\mathrm{GED}(G, G') = \min_{\{p_1,\ldots,p_n\} \in \mathcal{P}(G,G')} \sum_{i=1}^{n} \omega(p_i)$$

- We prefer counterfactuals that are **closer** to the original instance G

Oracle Related metrics

- **Oracle accuracy:**
  - Test accuracy of the oracle Φ
  - Low-accuracy oracles make any GCE questionable

- **Oracle calls:**
  - No. of Φ queries used to obtain G'
  - Latency-agnostic proxy for computational complexity and scalability.

Fidelity

- **Goal:** Measure how well a counterfactual G' reflects the trained oracle Φ's decision boundary.
- Assuming $\chi(G)$ measures if the original instance is classified correctly, fidelity is defined as :

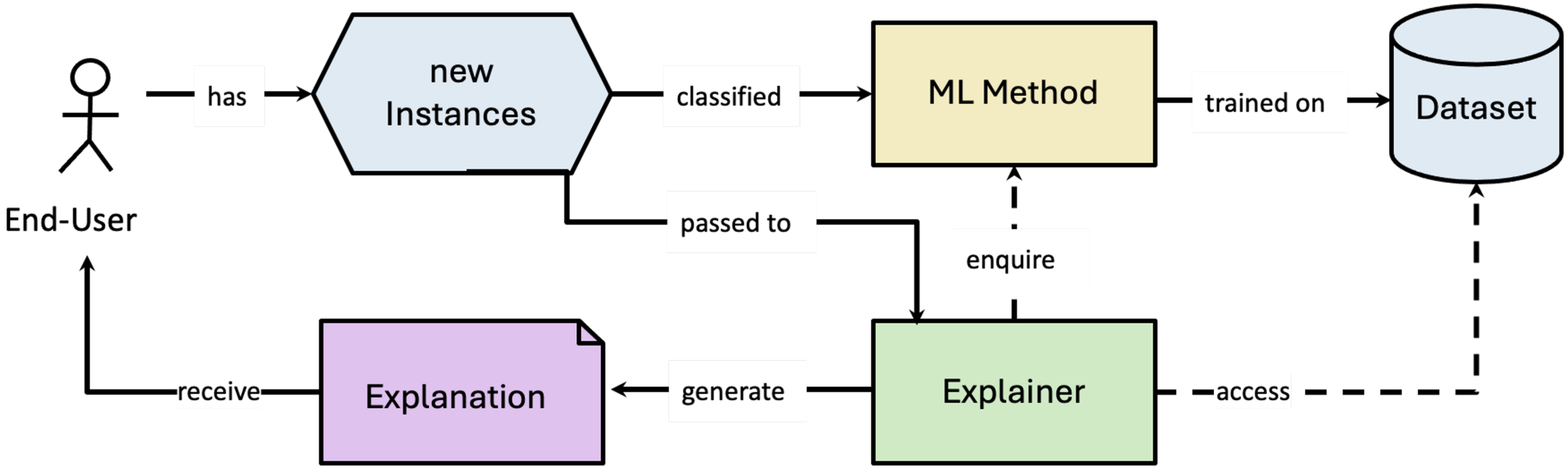
$$\chi(G) = \mathbb{I}\left[\Phi(G) = y\right]$$

$$\Psi(G, G') = \chi(G) - \mathbb{I}\left[\Phi(G') = y\right]$$

THE GRETEL FRAMEWORK

XAI WORKFLOW

- **A framework should implement all parts of the XAI workflow.**



GRETEL WHO?

Gretel is a **generic platform** that allows the researchers to **speed up** the process of **developing** *and* **testing** new
Graph Counterfactual Explanation Methods

- Object Oriented paradigm;
- Inversion of Control;
- Modular + Extensible;
- Reproducibility Ready

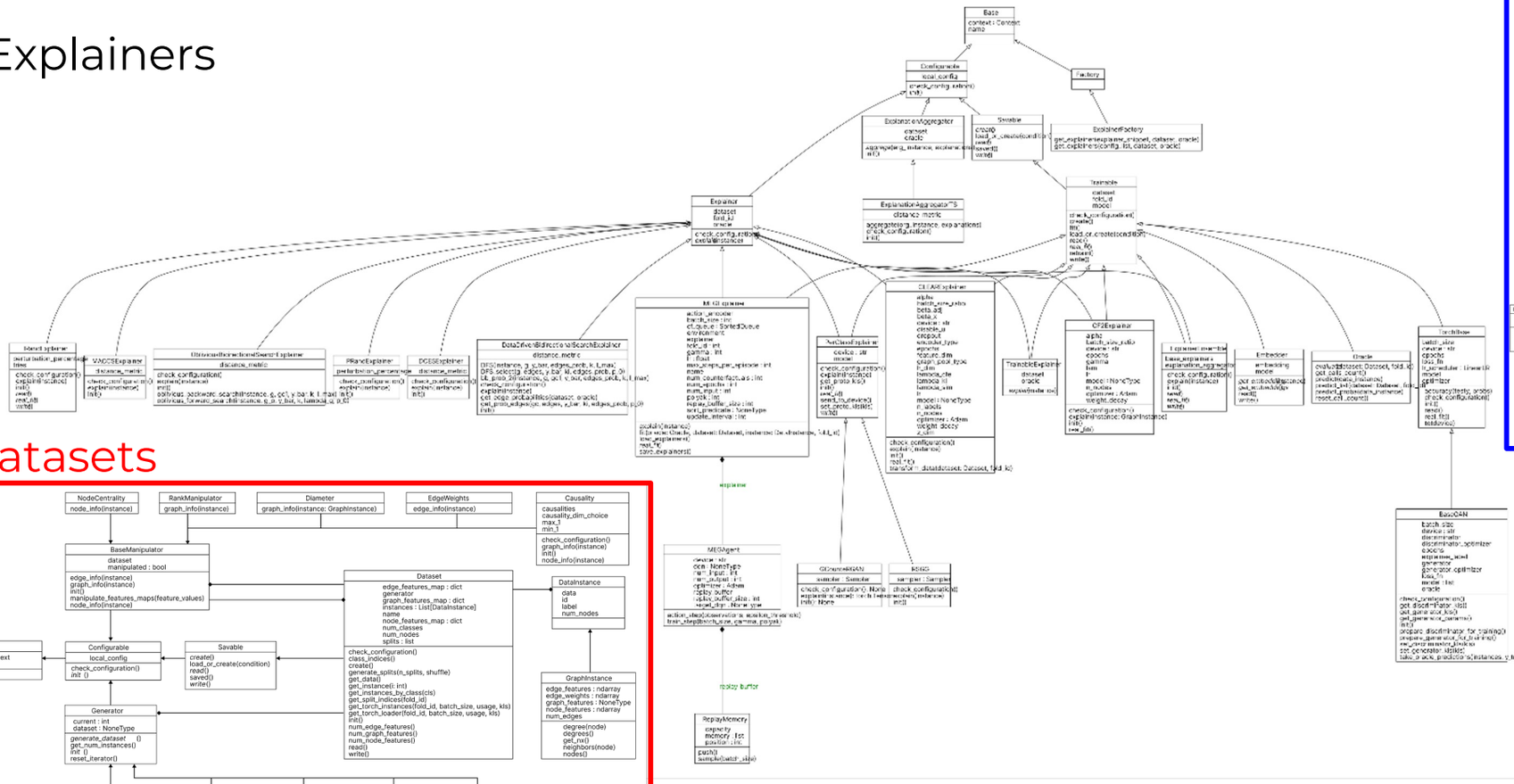
CIKM'22
(v1)

https://github.com/aiim-research/GRETEL
(v2)

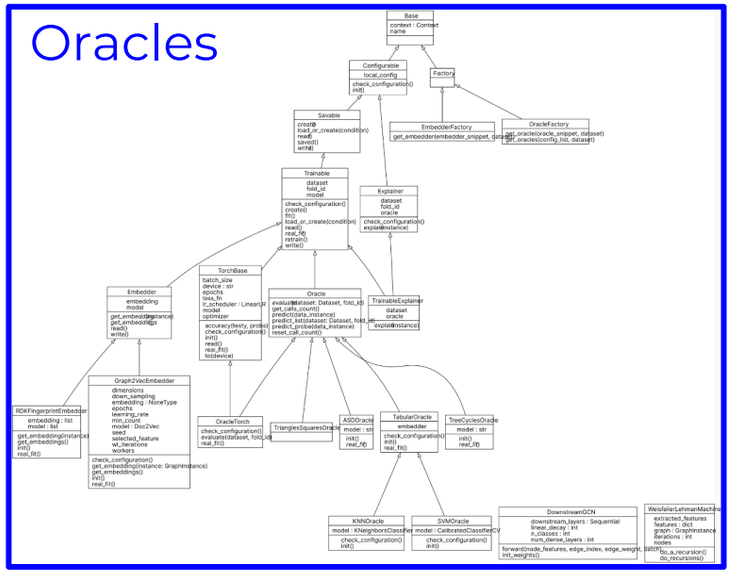
WSDM '23
(v1)

CAGING THE COMPLEXITY

Explainers



Oracles



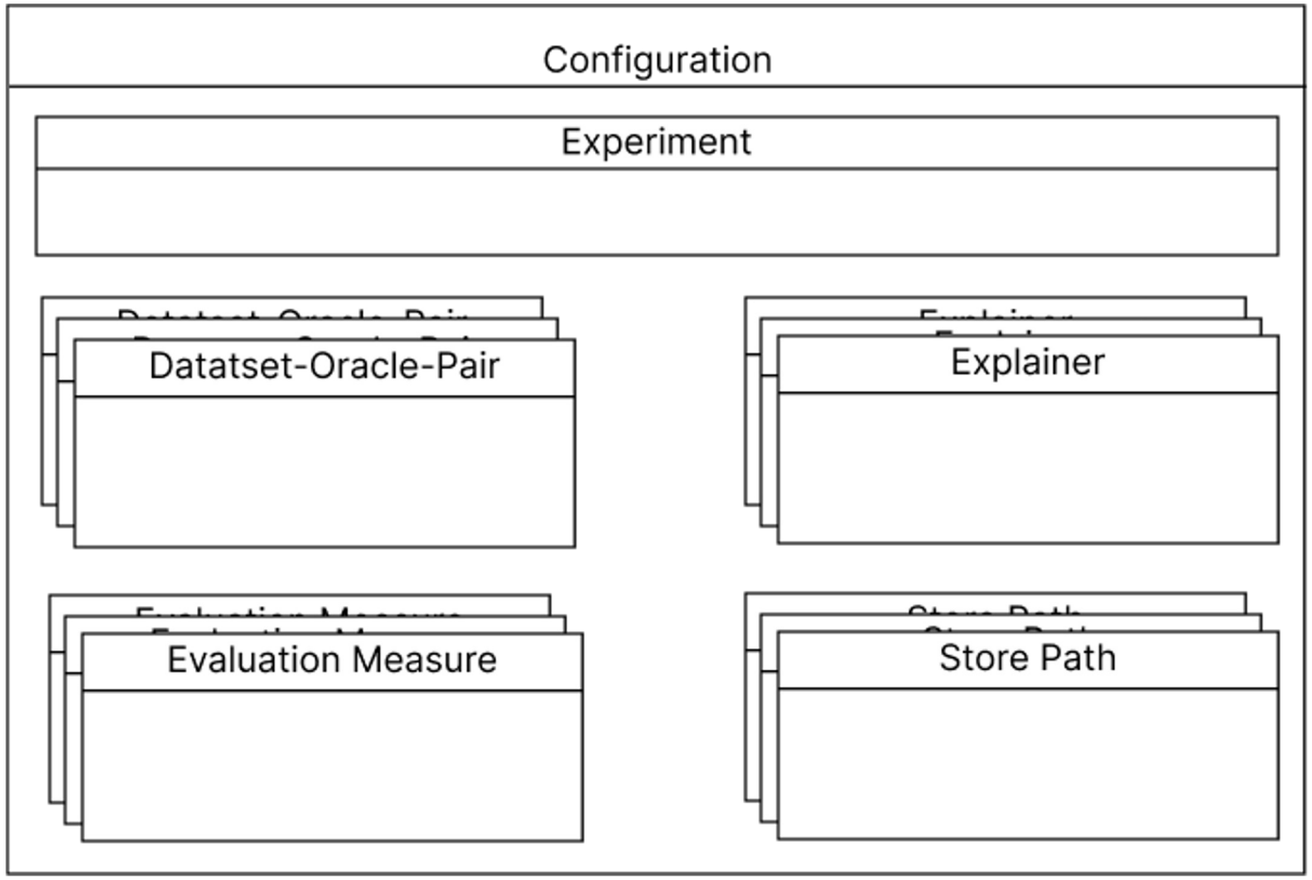
Datasets

Design Patterns (2)

- **Inversion of Control (IoC):** shifts control of object creation and execution to an external framework.

- **All experiments in GRETEL** are defined through a single, external configuration file.

Configuration Overview

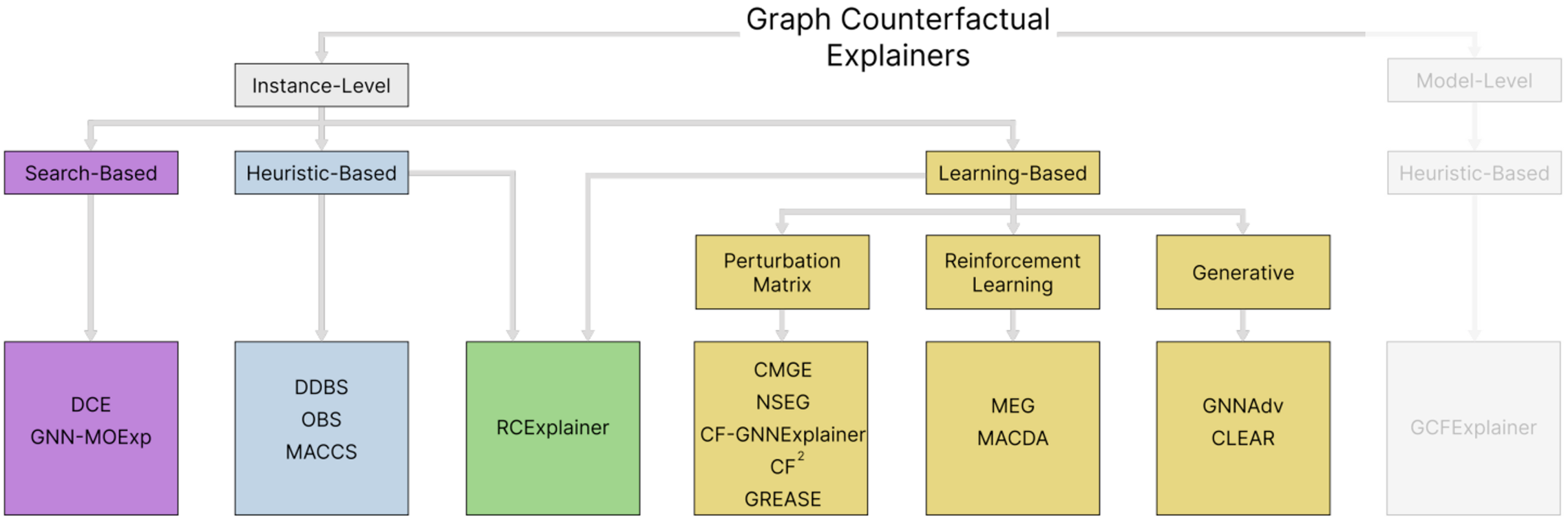



```
{
    "experiment" : {
        "scope": "examples_configs",
        "parameters" : {}
    },
    "do-pairs": [
        {"dataset" : { ... }, "oracle": { ... }},
        .
        {"dataset" : { ... }, "oracle": { ... }},
    ],
    "explainers": [
        { ... },
        .
        { ... }
    ],
    "evaluation_metrics": [
        { ... },

        .
        { ... }
    ],
    "store_paths": [
        { ... },

        .
        { ... }
    ]
}
```

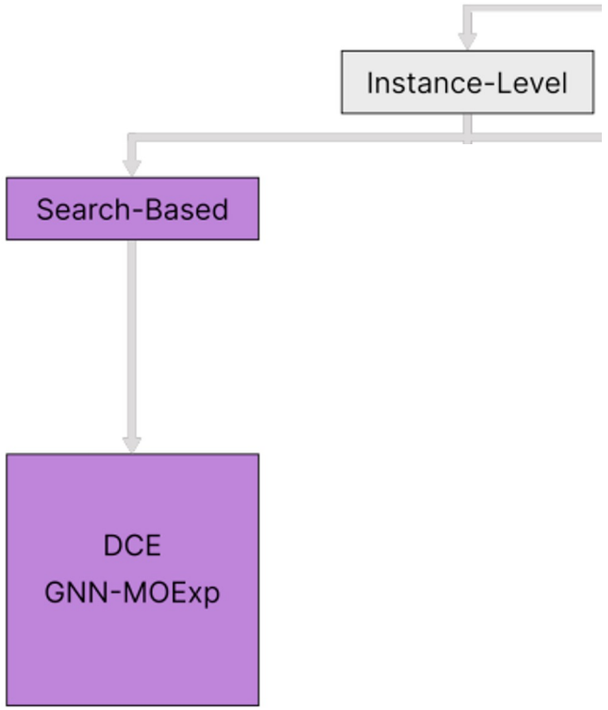
Graph Counterfactual Explanation Methods

Prado-Romero et al. “A survey on graph counterfactual explanations: definitions, methods, evaluation”, ACM CSUR 2023

GCE Methods Taxonomy

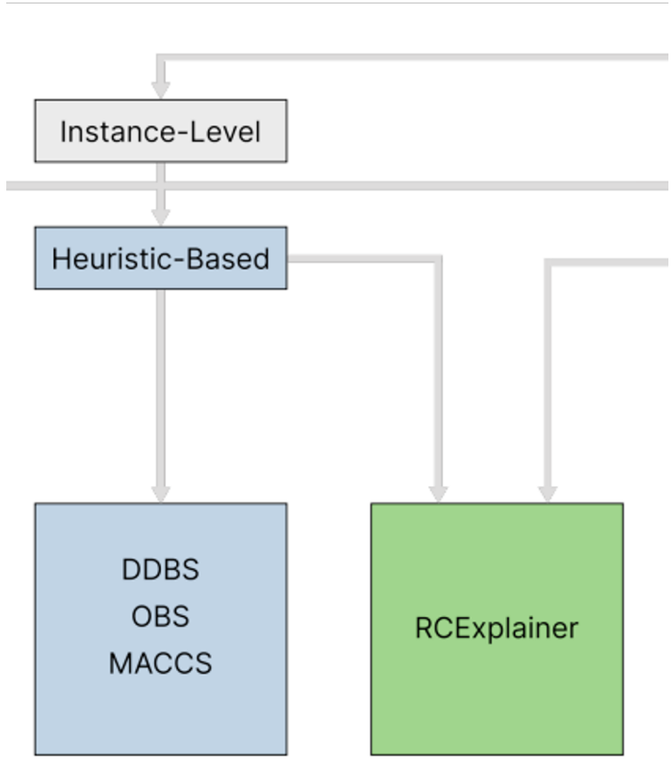


Search-Based GCE Methods



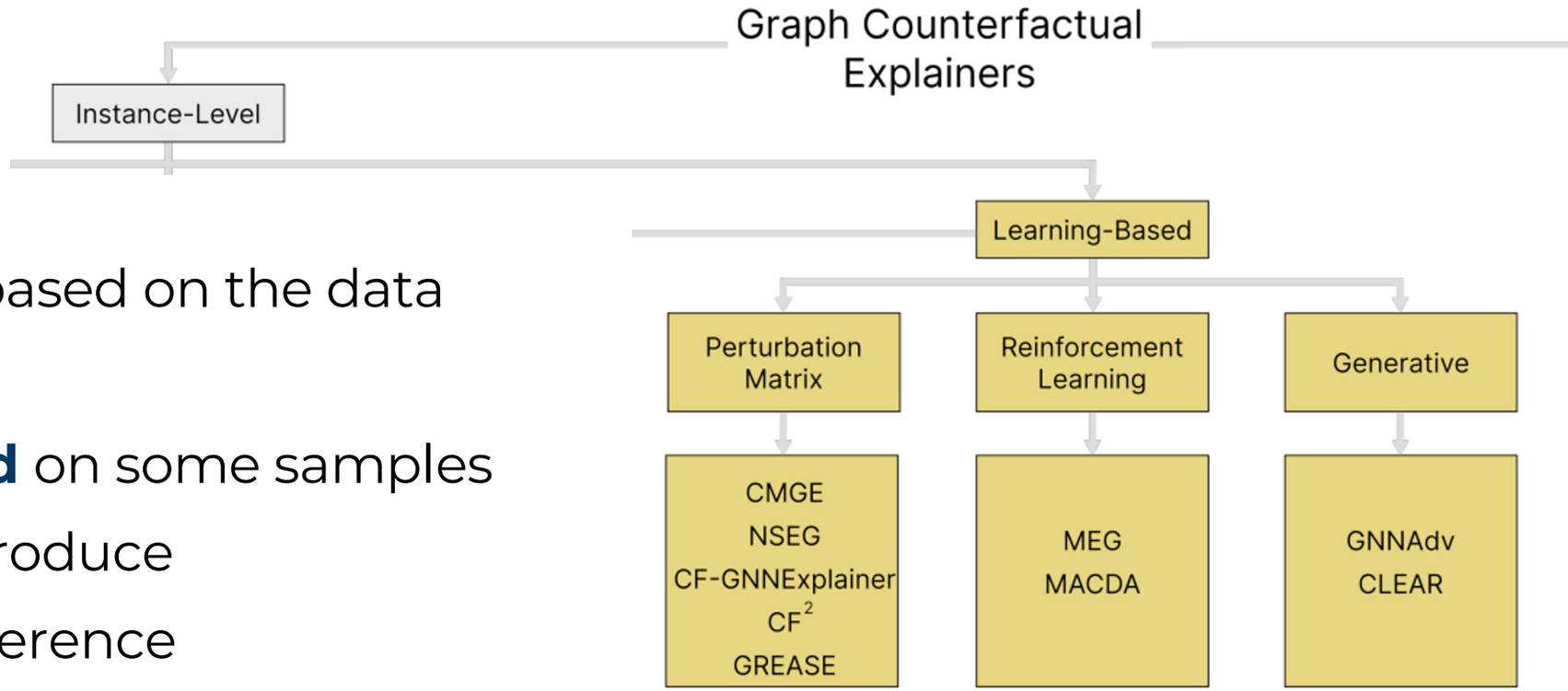
- Find a counterfactual **within the data**
- For a graph $\boldsymbol{G} \in \boldsymbol{\mathcal{G}}$ find a $\boldsymbol{G'} \in \boldsymbol{\mathcal{G}}$ s.t. $\boldsymbol{\Phi(G)} \neq \boldsymbol{\Phi(G')}$
- These methods **fail** to produce a counterfactual if the explainer **cannot access** the original **dataset**

Heuristic-Based GCE Methods



- **Perturb** the original graph such that $\Phi(G) \neq \Phi(G')$ **without** accessing the original dataset
- **Requires** to define the perturbation **rules** after a **careful examination** of the data

Learning-Based GCE Methods

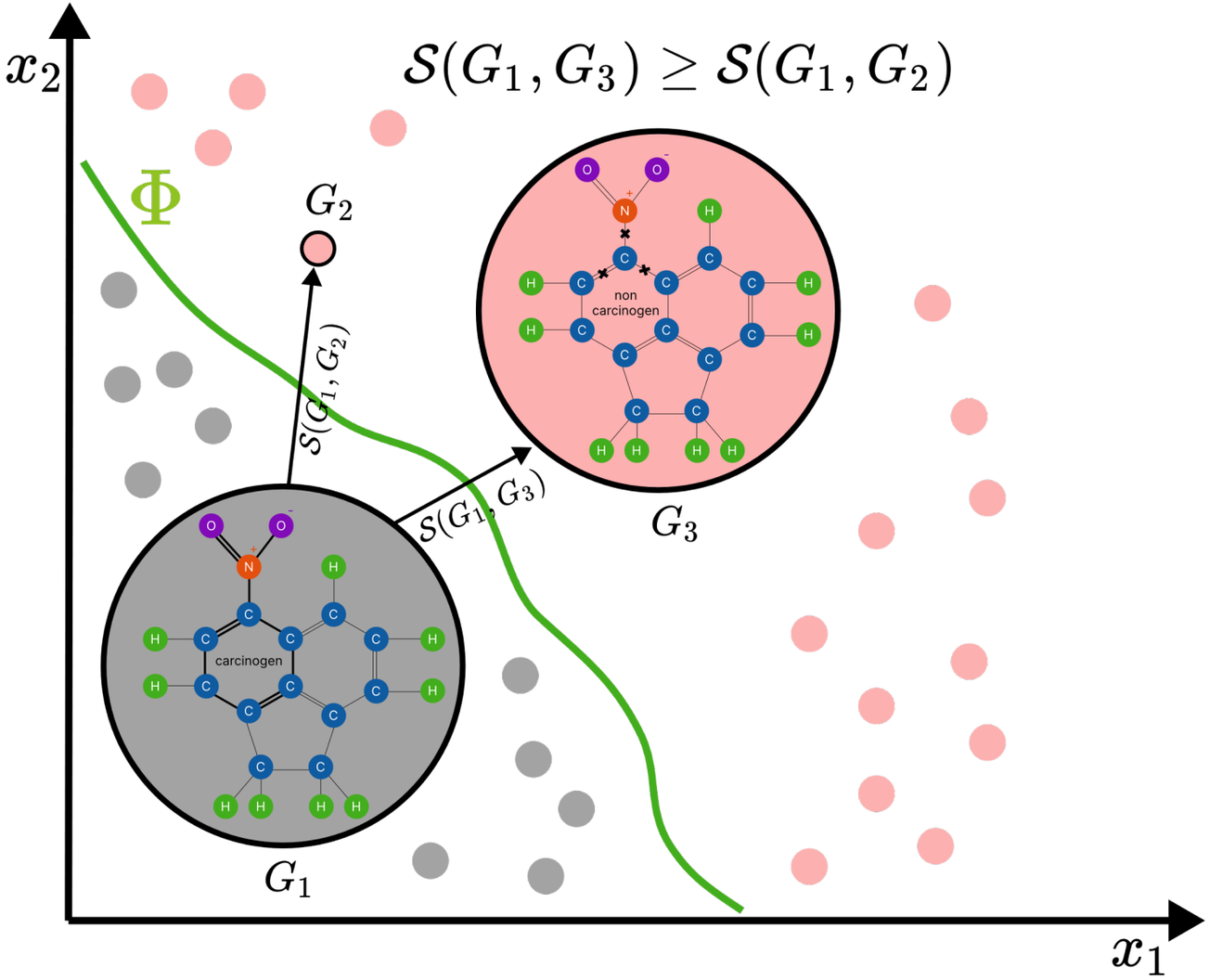


- **Learn the heuristic** based on the data
- Explainers are **trained** on some samples and can be used to produce counterfactuals at inference

BASELINE GRAPH COUNTERFACTUAL EXPLAINER

L. Faber, A. K. Moghaddam, and R. Wattenhofer. 2020. Contrastive Graph Neural Network Explanation. In Proc. of the 37th Graph Repr. Learning and Beyond Workshop at ICML 2020. Int. Conf. on Machine Learning, 28

DCE (Distribution Complaint Explanations)



DCE

$$G^* = \arg \min_{G' \in \mathcal{G}, \Phi(G) \neq \Phi(G')} d(G, G')$$

HEURISTIC-BASED EXPLAINERS

Abrate C, Bonchi F. Counterfactual graphs for explainable classification of brain networks. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining 2021 Aug 14 (pp. 2495-2504).

Wellawatte GP, Gandhi HA, Seshadri A, White AD. A Perspective on Explanations of Molecular Prediction Models. Journal of Chemical Theory and Computation. 2023 Mar 27;19(8):2149-60.

OBS & DDBS

Oblivious and **D**ata-**D**riven **B**idirectional **S**earch

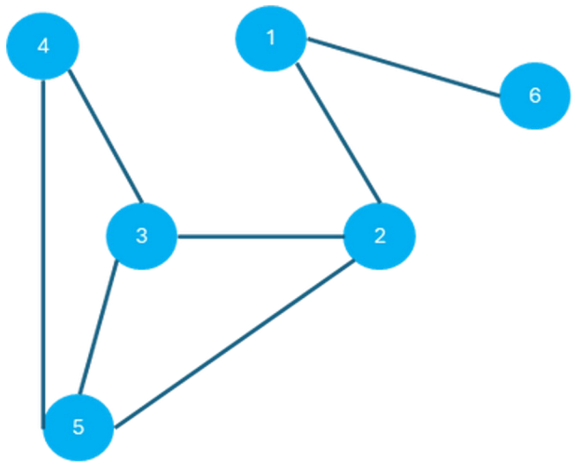
Oblivious Bidirectional Search (OBS)

- **Select a graph instance to explain:** The method is designed for explaining graph classifiers

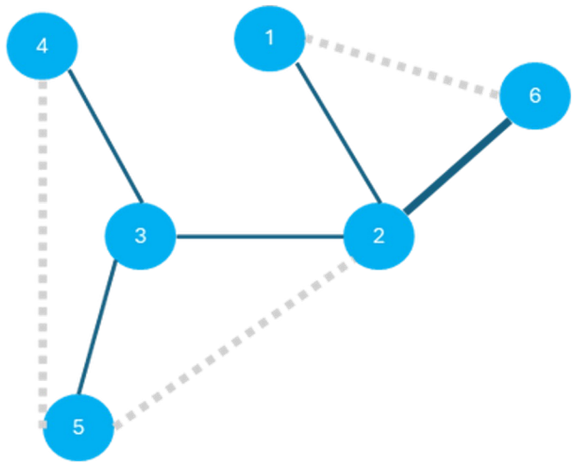
- **Perturb edges until a counterfactual is reached:** Randomly modify the edges of the graph x until the prediction of the model for the perturbed instance x' is different to the prediction for the original instance

- **Reduce the distance between x and x':** Try to undo as much as possible the changes made to obtain the x' while keeping its predicted label different from that of x

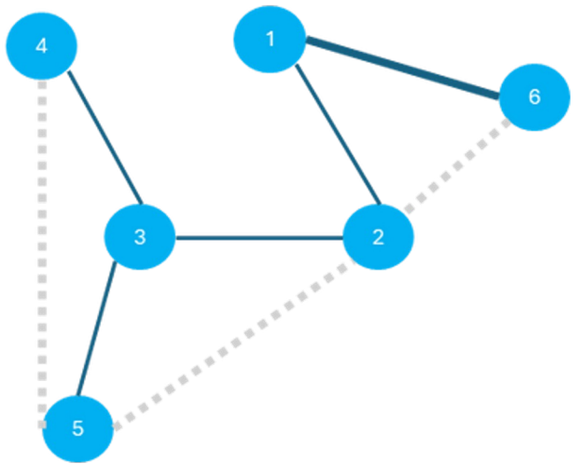
Oblivious Bidirectional Search (OBS)



Cyclic Graph



Step 1: Find a Counterfactual



Step 2: Reduce distance between original graph and counterfactual

Data-Driven Bidirectional Search

- **Select a graph instance to explain**. The method is designed for explaining graph classifiers.

- **Sort edges** according to their frequency of appearance in each class

- **Perturb edges until a counterfactual is reached**. Considering the order of the edges for the counterfactual class, modify the edges of the graph x until the prediction of the model for the perturbed instance x' is different to the prediction for the original instance

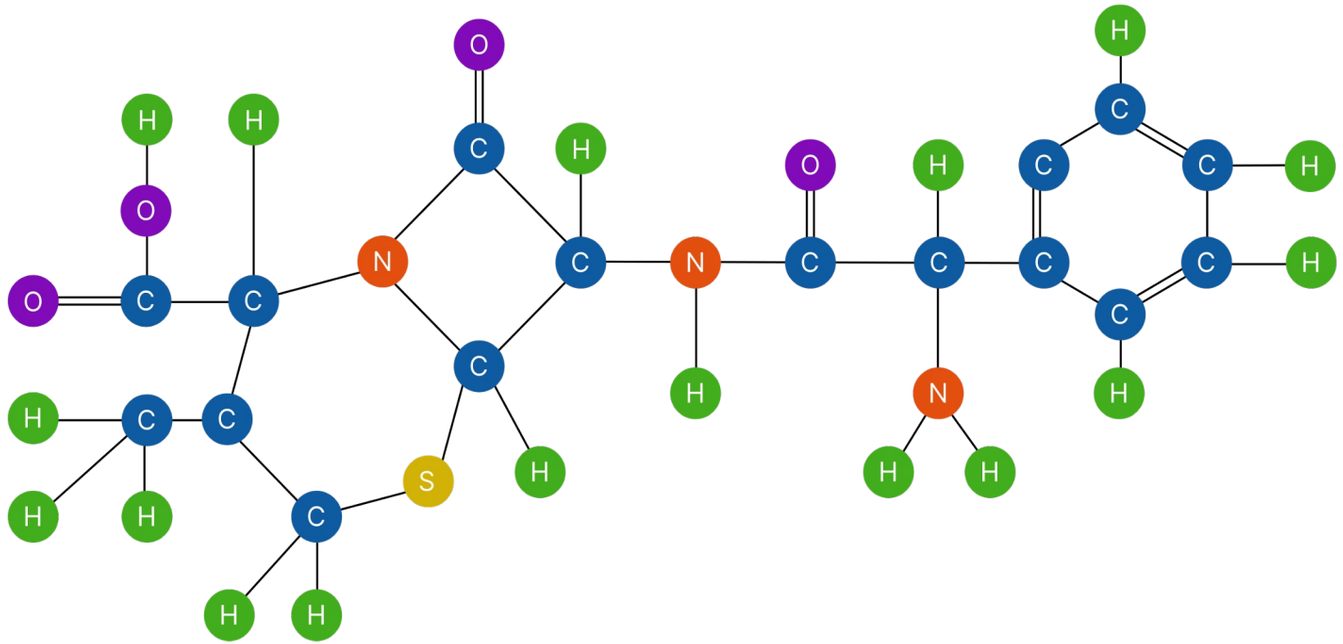
- **Reduce the distance between x and x'**. Try to undo as much as possible the changes made to obtain the x' while keeping its predicted label different from that of x

MACCS
Model Agnostic Counterfactual Compounds with STONED[1]

1. Superfast Traversal, Optimization, Novelty, Exploration and Discovery (STONED) method for exploration of chemical space

MACCS

Uses SMILES representations of molecules



CC1=C(N2C(C(C2=O)NC(=O)C(C3=CC=CC=C3)N)SC1)C(=O)O

MACCS

- Expand the chemical space around the original molecule
- Select some molecules from the expansion space via **clustering**
- Choose multiple ones **closest** to the input

MACCS (EXAMPLE)

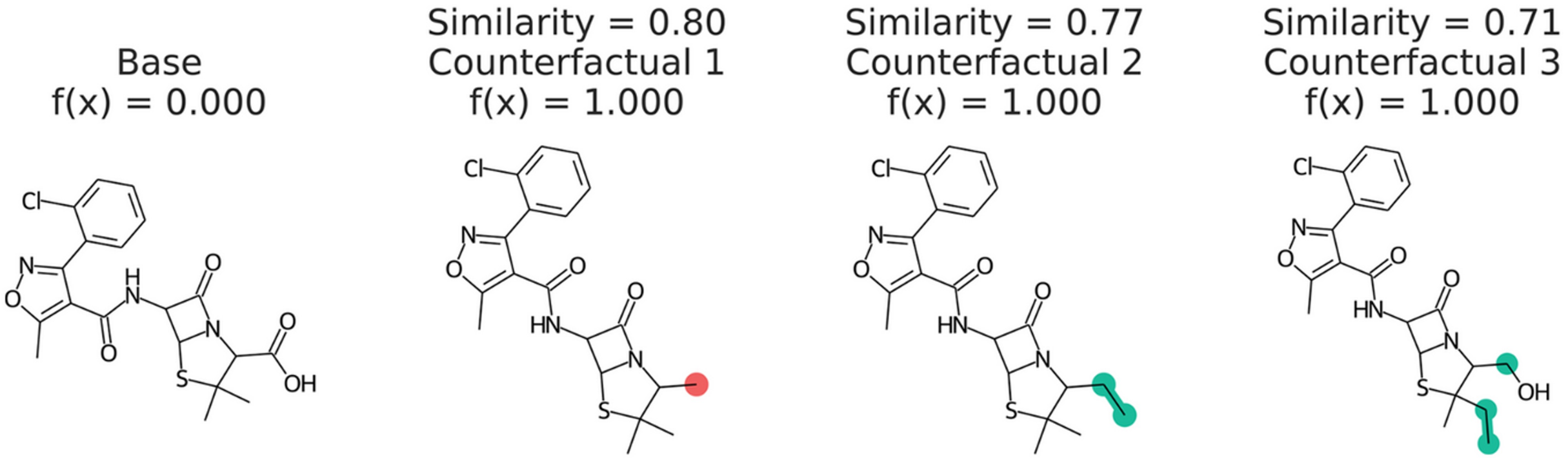


Figure 1. Counterfactuals of a molecule which cannot permeate the blood–brain barrier. Similarity is the Tanimoto similarity of ECFP4 fingerprints.[130] Red indicates deletions and green indicates substitutions and addition of atoms. Republished from ref 9 with permission from the Royal Society of Chemistry. Copyright 2022.

LEARNING-BASED EXPLAINERS

D. Numeroso and D. Bacciu. 2021. Meg: Generating molecular counterfactual explanations for deep graph networks. In 2021 Int. Joint Conf. on Neural Networks. IEEE, 1–8

Wellawatte GP, Gandhi HA, Seshadri A, White AD. A Perspective on Explanations of Molecular Prediction Models. Journal of Chemical Theory and Computation. 2023 Mar 27;19(8):2149-60.

Tan, J., Geng, S., Fu, Z., Ge, Y., Xu, S., Li, Y. and Zhang, Y., 2022, April. Learning and evaluating graph neural network explanations based on counterfactual and factual reasoning. In Proceedings of the ACM Web Conference 2022 (pp. 1018-1027).

Ma, J., Guo, R., Mishra, S., Zhang, A. and Li, J., 2022. Clear: Generative counterfactual explanations on graphs. Advances in Neural Information Processing Systems, 35, pp.25895-25907.

Prado-Romero MA, Prenkaj B, Stilo G. Robust Stochastic Graph Generator for Counterfactual Explanations. AAAI'24

Prenkaj B, Zaradoukas E, Kasneci G. Graph Inverse Style Transfer for Counterfactual Explanations. ICML'25

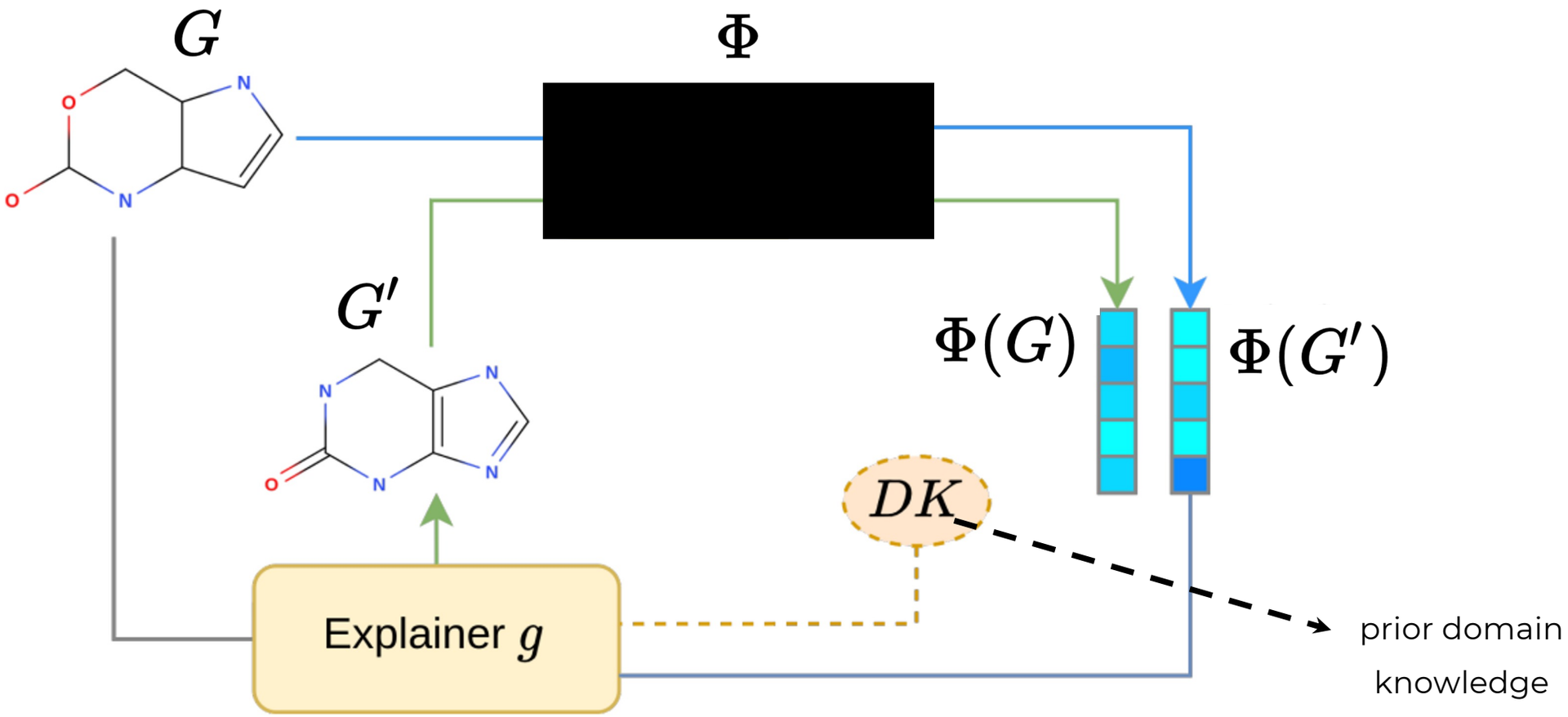
MEG

Molecular **E**xplanation **G**enerator

MEG

- **Reinforcement Learning-based Explainer**. It is specifically designed for the molecular domain, however, can be adapted to other domains.
- **The initial state is the original instance to be explained**. From there the RL-Agent will learn to modify it to generate a counterfactual
- **The set of actions considers domain knowledge**. It is based on the MolDQN model to ensure the atom and bonds additions/removals produce valid molecules
- **The multi-objective reward** function exploits the prediction of the model Φ

MEG (ARCHITECTURE)



MEG

- **Goodness of an action-state pair**. It is evaluated by a function Q that is approximated using a deep network

- **Policy**. It is a function outputting the best possible action; uses a decaying ϵ-greedy

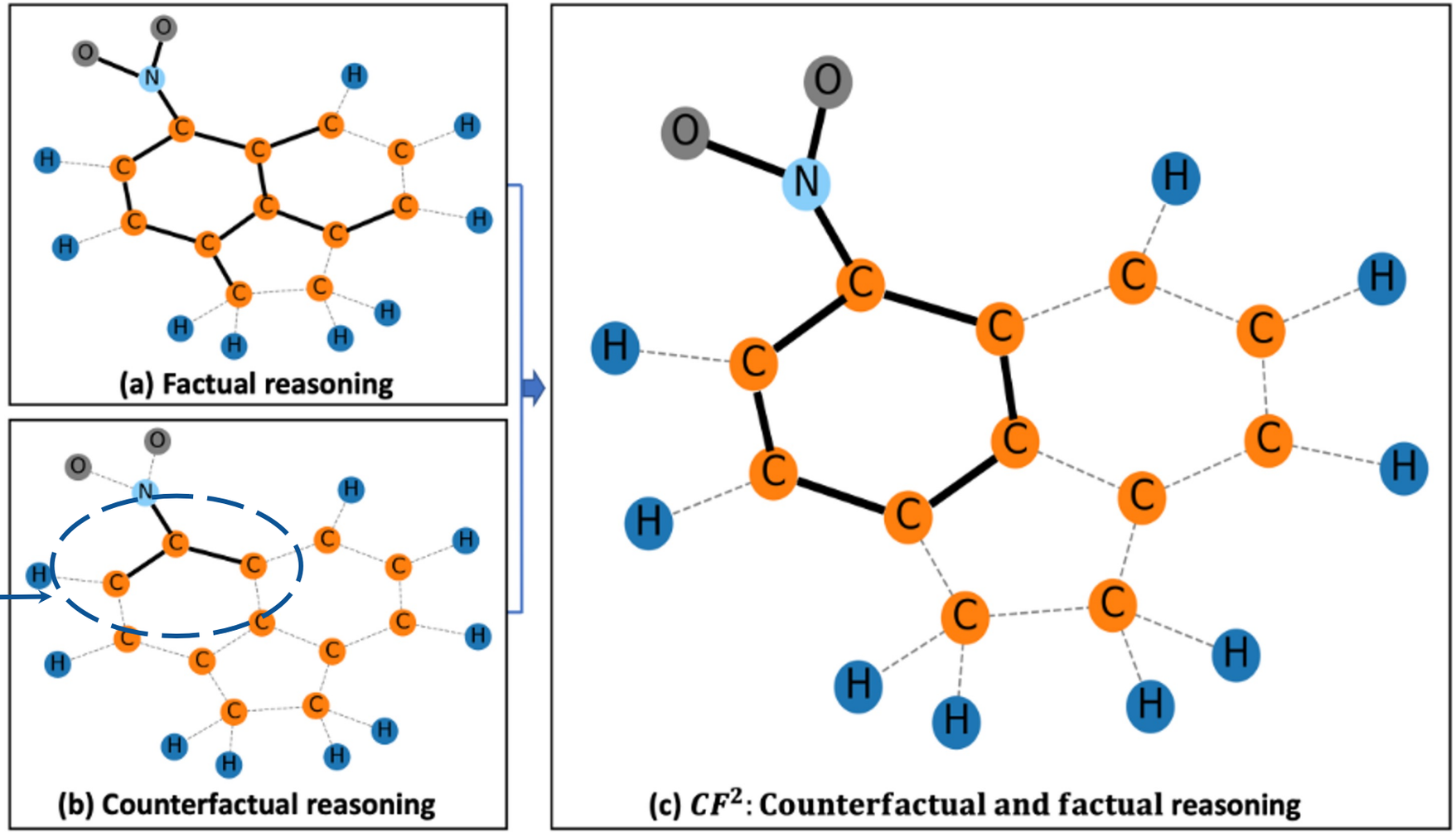
- **The training is performed individually for each instance**

CF^2

Counter**F**actual and **F**actual explanations

CF²

these edges, if removed, create the counterfactual



CF² (Necessity & Sufficiency)

An explanation needs to be **necessary** and **sufficient**

Counterfactual Reasoning

Factual Reasoning

CF² (Sufficiency)

Factual Reasoning

$$\Phi(A_* \cdot M, X_* \cdot F) = y$$

Masked adjacency matrix

Masked node feature vectors

Both masks are learned by the explainer

CF² (Relaxed Optimization)

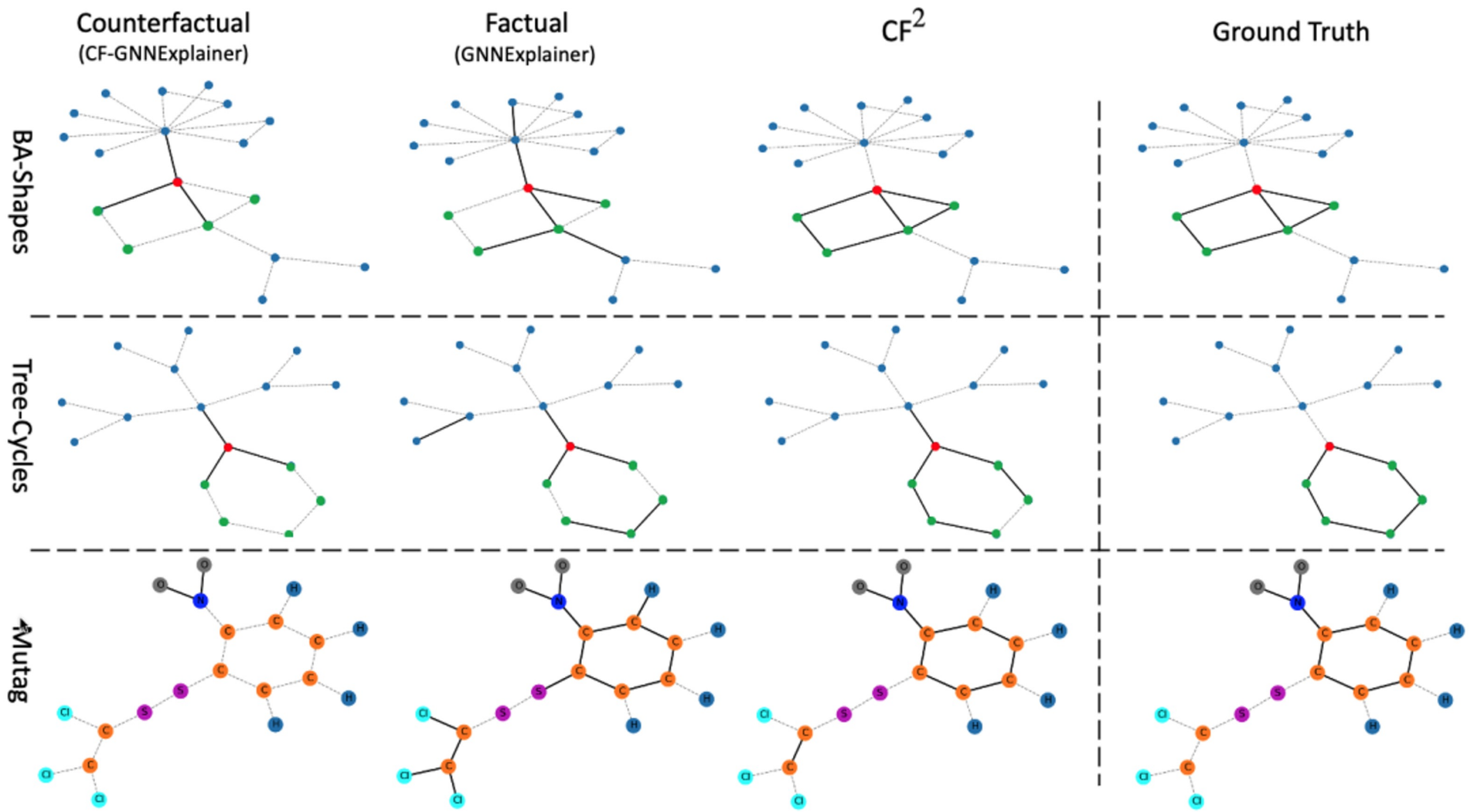
$$L_f = \mathbf{ReLU}(\gamma + P(\Phi(A * M, X * F) = \neg y) - S_f(M, F))$$

$$L_c = \mathbf{ReLU}(\gamma - P(\Phi(A - A * M, X - X * F) = \neg y) - S_c(M, F))$$

} Relax the constraints into a pairwise contrastive loss

$$\mathbf{minimize}\ ||M||_1 + ||F||_1 + \lambda(\alpha L_f + (1 - \alpha)L_c)$$

CF² (Qualitative Results)



CLEAR

Generative **C**ounterfactua**L** **E**xpl**A**nation gene**R**ator for graphs

CLEAR

First work to treat **causality** when producing counterfactuals even if underexplored

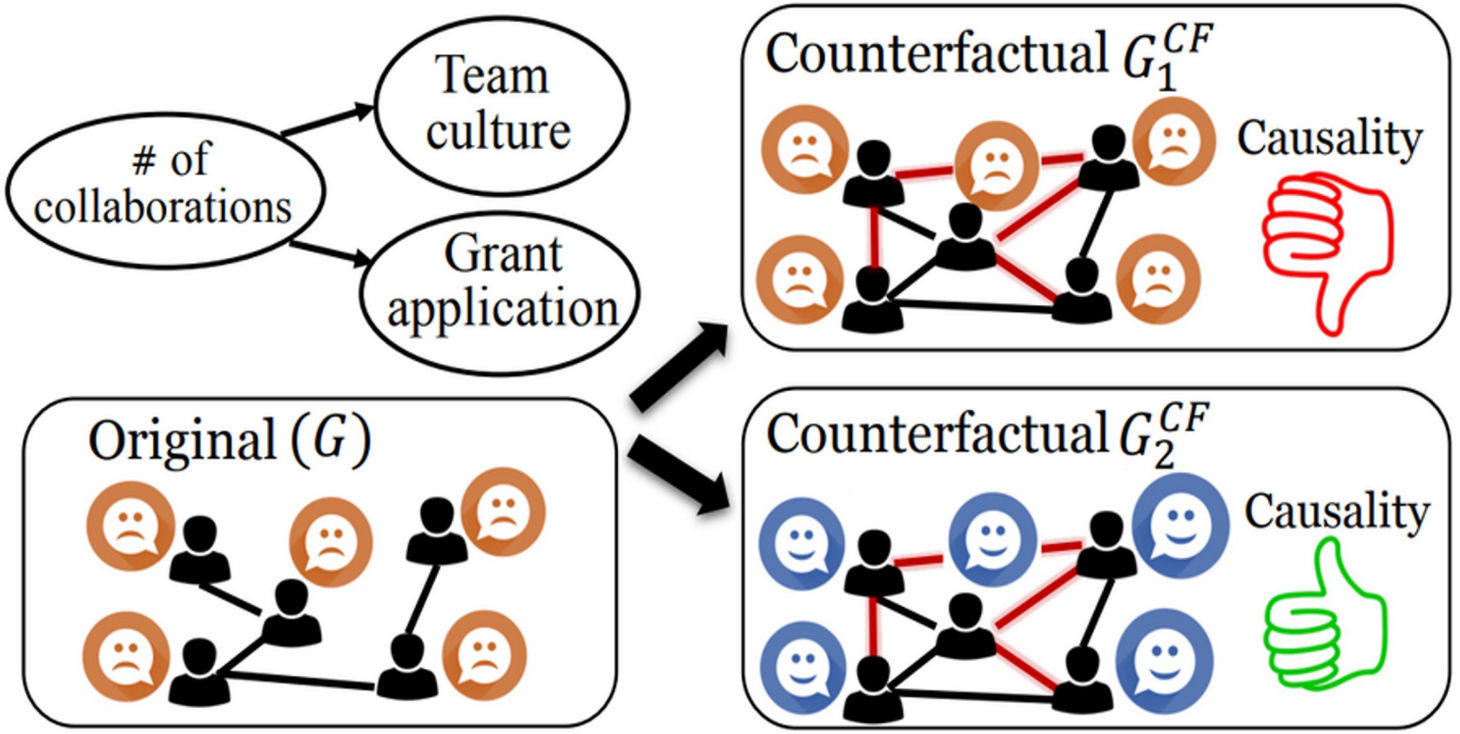


Image taken from: Ma J, Guo R, Mishra S, Zhang A, Li J. Clear: Generative counterfactual explanations on graphs. Advances in Neural Information Processing Systems. 2022 Dec 6;35:25895-907.

CLEAR

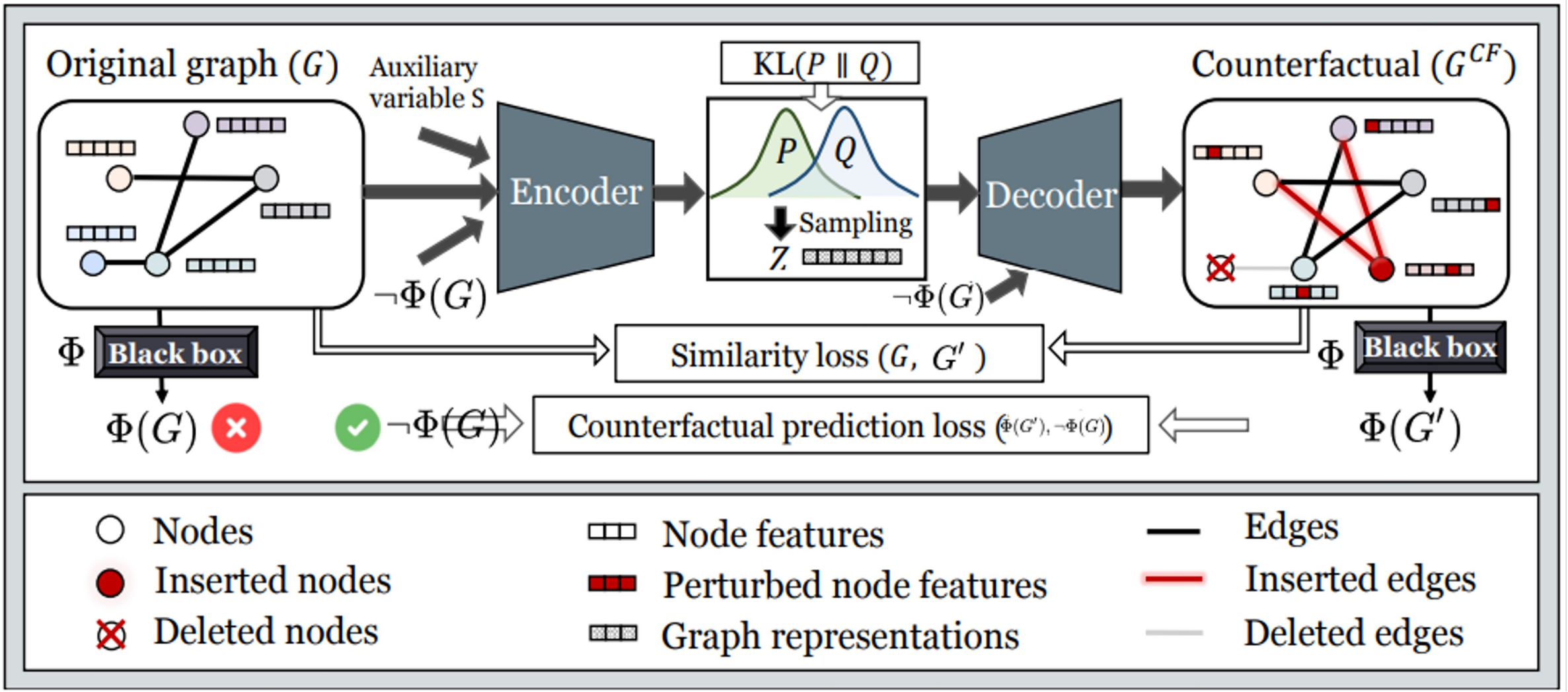
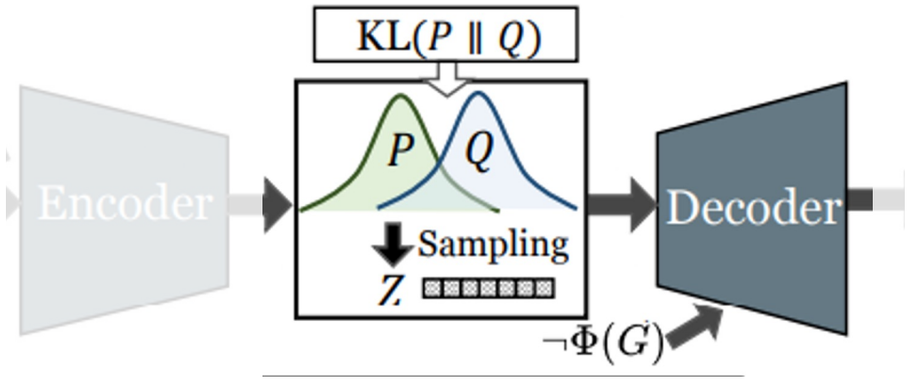


Image taken from: Ma J, Guo R, Mishra S, Zhang A, Li J. Clear: Generative counterfactual explanations on graphs. Advances in Neural Information Processing Systems. 2022 Dec 6;35:25895-907.

CLEAR

- One can generate **multiple counterfactuals** from sampling multiple Z
- The decoder produces a probabilistic graph where edges have weights $\mathbb{R}_0^1$
- Binarize the graph according to the Bernoulli distribution



CLEAR

$$d(G, G') = d_A(A, \text{Bernoulli}(A')) + d_X(X, X')$$

distance between the original adjacency matrix and the generated one

distance between the original node features and the generated ones

RSGG-CE

Robust **S**tochastic **G**raph **G**enerator for **C**ounterfactual **E**xplanations

RSGG-CE

- **Cornerstone paper** in the debate "*Are generative counterfactual explanation approaches worth it?*"
- **Besides CLEAR, all other explainers are discriminative**
- **Uses Residual GANs to learn how to generate counterfactuals**

RSGG-CE (TRAINING)

We need to integrate the decision of the oracle to guide the generation of plausible counterfactuals

$$\mathbb{I}[\Phi(A) \neq c] = \begin{cases} 1 & \text{if } \Phi(A) \neq c \\ 0 & \text{otherwise} \end{cases}$$

We'll use this indicator function in the discriminator to induce the generator to correctly generate counterfactuals

RSGG-CE (TRAINING)

what are the instances belonging to class c?

$$\mathcal{A}_c = \{A \mid A \in \mathcal{A} \ \wedge \neg \mathbb{I}[\Phi(A) \neq c]\}$$

what are the instances **NOT** belonging to class c?

$$\mathcal{A}_{\neg c} = \{A \mid A \in \mathcal{A} \ \wedge \mathbb{I}[\Phi(A) \neq c]\}$$

RSGG-CE (TRAINING)

$$\mathcal{L}(\mathbb{G},\mathbb{D}) = \mathop{\mathbb{E}}_{A\in\mathcal{A}}\Big[\log \mathbb{D}(A)\Big] + \mathop{\mathbb{E}}_{A_z\in\mathcal{A}}\Big[\log(1-\mathbb{D}\big(A_z+\mathbb{G}(A_z)\big))\Big]$$

$$\mathcal{L}_{\Phi,c}(\mathbb{G},\mathbb{D}) = \overbrace{\sum_{A\in\mathcal{A}_{\neg c}} \log \mathbb{D}(A)}^{\mathbb{D}\text{'s optimization on real data}} + \overbrace{\sum_{A\in\mathcal{A}_c, A_g = A+\mathbb{G}(A)} \mathbb{I}\big[\Phi(A_g)\neq c\big]\log\mathbb{D}(A_g) + \underbrace{\sum_{A\in\mathcal{A}_c}\log\big(1-\mathbb{D}(A+\mathbb{G}(A))\big)}_{\mathbb{G}\text{'s optimization}}}^{\mathbb{D}\text{'s optimization on generated data}}$$

RSGG-CE (TRAINING)

$$\mathcal{L}(\mathbb{G},\mathbb{D}) = \mathop{\mathbb{E}}_{A\in\mathcal{A}}\Big[\log \mathbb{D}(A)\Big] + \mathop{\mathbb{E}}_{A_z\in\mathcal{A}}\Big[\log(1-\mathbb{D}(A_z+\mathbb{G}(A_z)))\Big]$$

$$\mathcal{L}_{\Phi,c}(\mathbb{G},\mathbb{D}) = \overbrace{\sum_{A\in\mathcal{A}_{\neg c}} \log \mathbb{D}(A)}^{\mathbb{D}\text{'s optimization on real data}} + \overbrace{\sum_{A\in\mathcal{A}_c, A_g = A+\mathbb{G}(A)} \mathbb{I}\big[\Phi(A_g)\neq c\big]\log \mathbb{D}(A_g)}^{\mathbb{D}\text{'s optimization on generated data}}$$

$$\underbrace{+\sum_{A\in\mathcal{A}_c} \log\left(1-\mathbb{D}(A+\mathbb{G}(A))\right)}_{\mathbb{G}\text{'s optimization}}$$

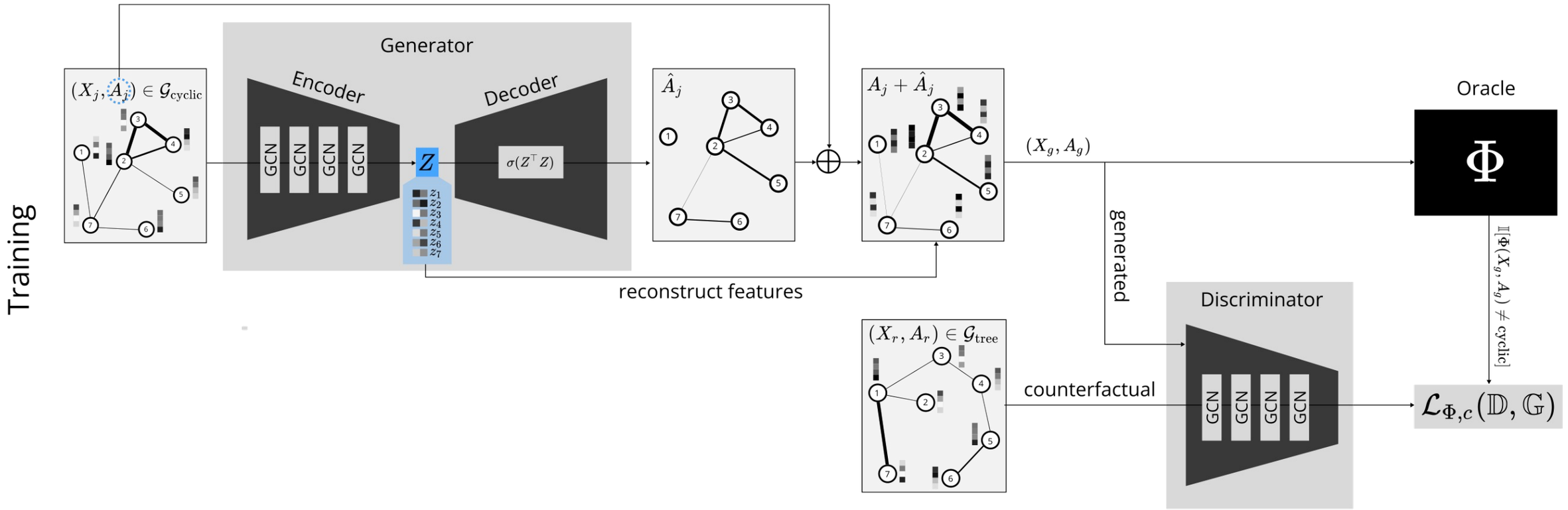
RSGG-CE (TRAINING)

$$\mathcal{L}(\mathbb{G}, \mathbb{D}) = \mathop{\mathbb{E}}_{A \in \mathcal{A}} \Big[\log \mathbb{D}(A) \Big] + \mathop{\mathbb{E}}_{A_z \in \mathcal{A}} \Big[\log(1 - \mathbb{D}\big(A_z + \mathbb{G}(A_z)\big)) \Big]$$

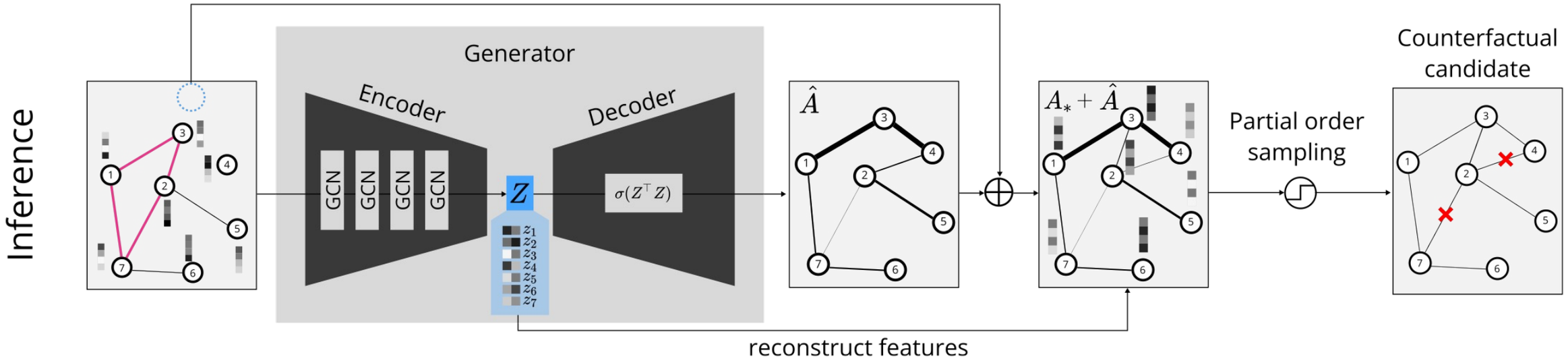
$$\mathcal{L}_{\Phi,c}(\mathbb{G}, \mathbb{D}) = \overbrace{\sum_{A \in \mathcal{A}_{\neg c}} \log \mathbb{D}(A)}^{\mathbb{D}\text{'s optimization on real data}} + \overbrace{\sum_{A \in \mathcal{A}_c, A_g = A + \mathbb{G}(A)} \mathbb{I}\big[\Phi(A_g) \neq c\big] \log \mathbb{D}(A_g)}^{\mathbb{D}\text{'s optimization on generated data}}$$

$$\underbrace{+ \sum_{A \in \mathcal{A}_c} \log \big(1 - \mathbb{D}(A + \mathbb{G}(A))\big)}_{\mathbb{G}\text{'s optimization}}$$

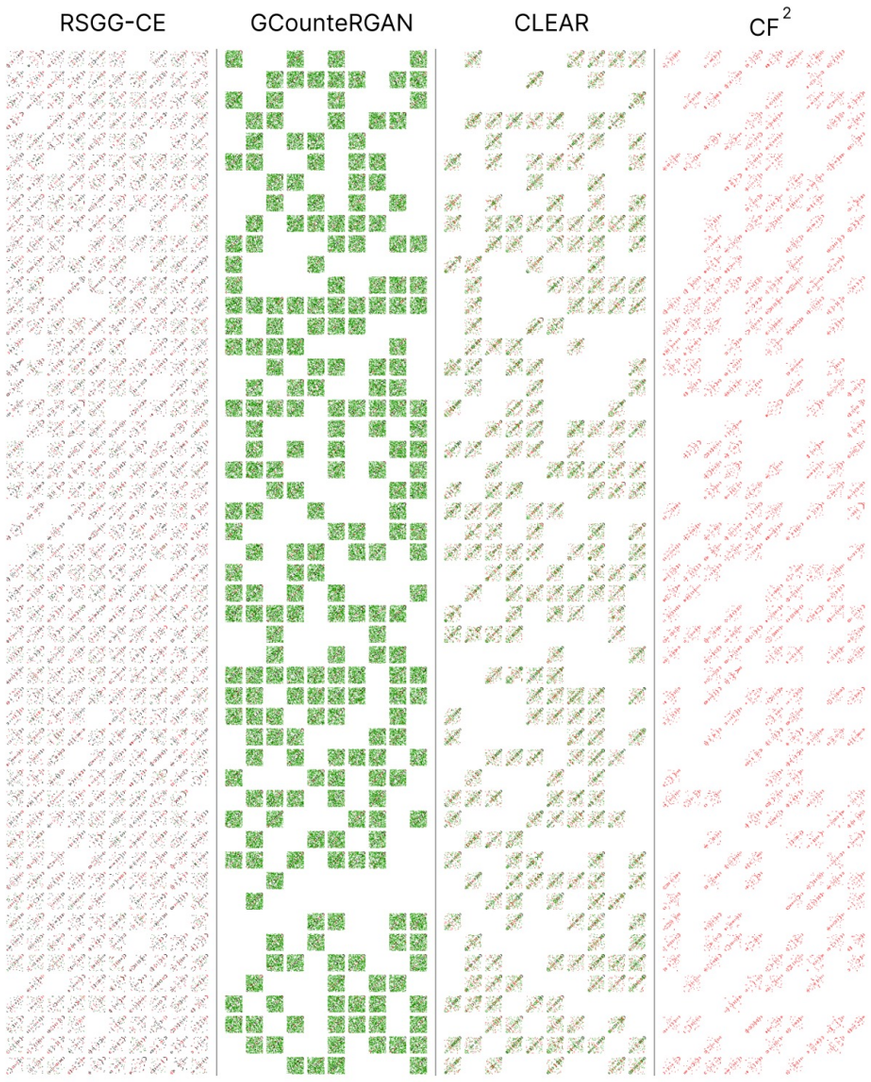
RSGG-CE (TRAINING)

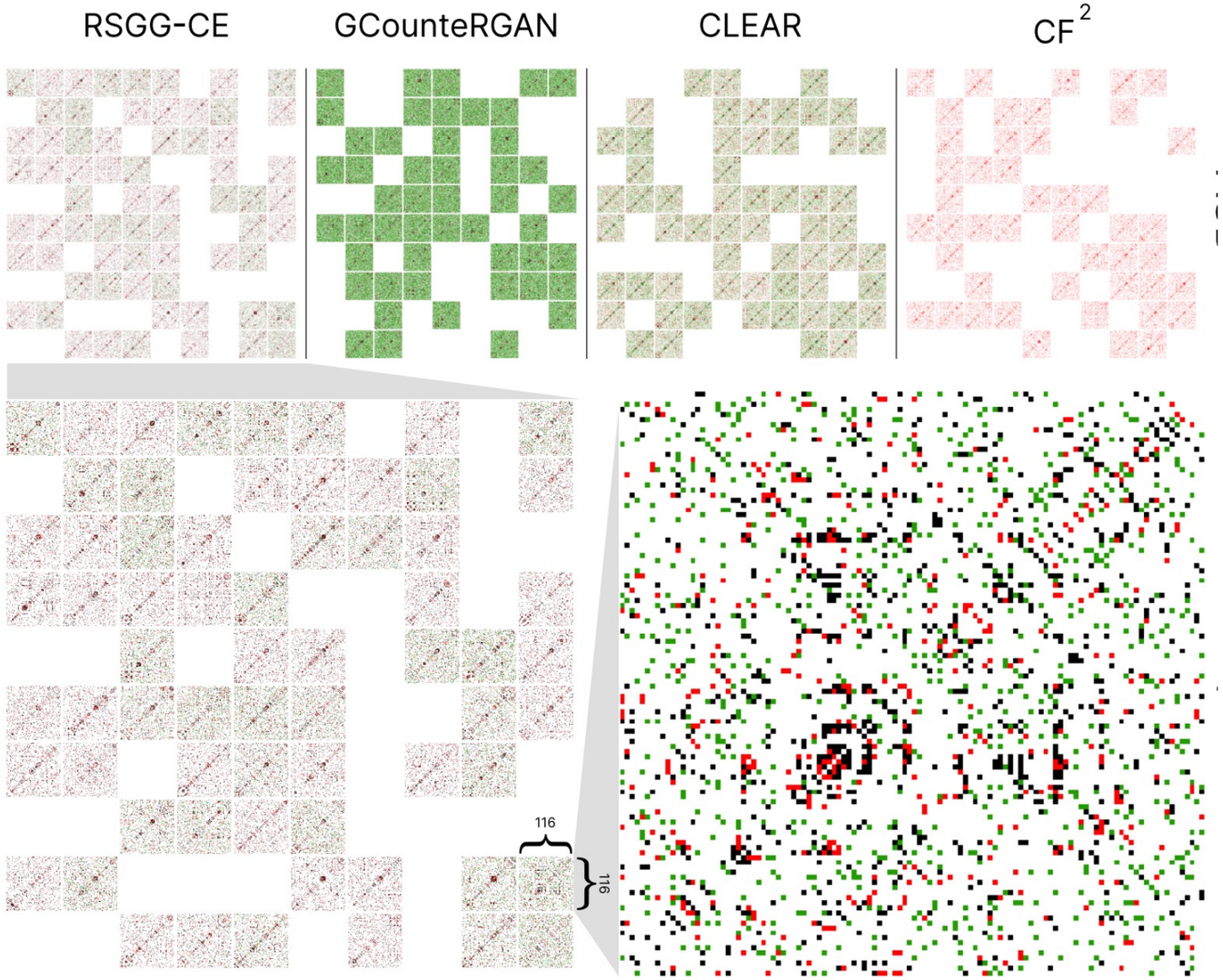


RSGG-CE (INFERENCE)



RSGG-CE (RECAP)





Tree-Cycle

ASD

RSGG-CE (RECAP)

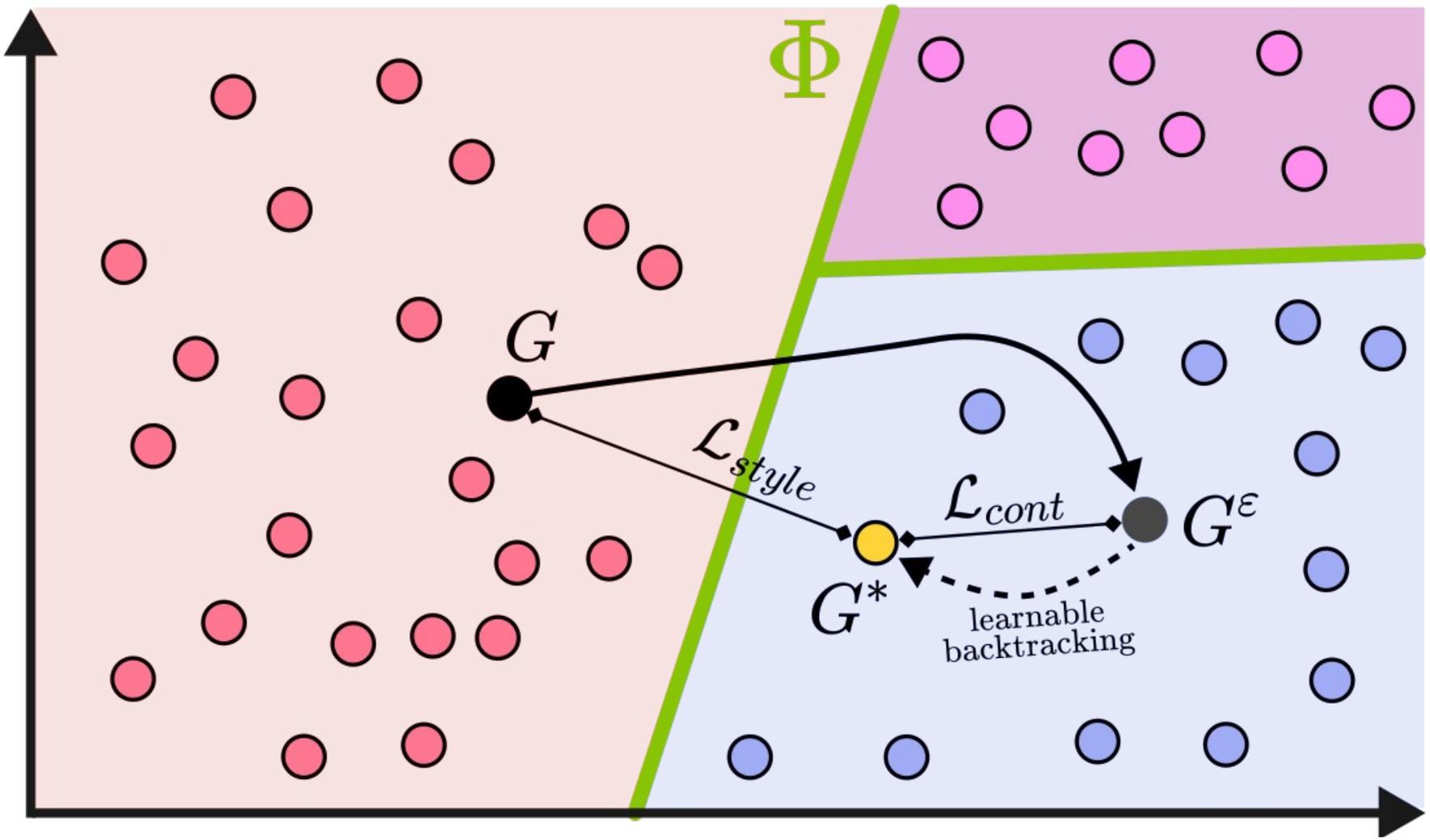
- We train the generator on graphs of class c (**the class we want to explain**)

- We train the discriminator on graphs different from class c and the synthetic data generated

- Because the generator needs to fool the discriminator, it'll learn to produce graphs of class **not c**

GIST
Graph Inverse Style Transfer for Counterfactual Explainability

GIST


Φ
G
$\mathcal{L}_{style}$
$\mathcal{L}_{cont}$
G^{ϵ}
G^{*}
learnable
backtracking

GIST (GRAPH STYLE AND CONTENT)

$$G = (X, A) \begin{cases} X \in \mathbb{R}^{n \times d} \\ A \in \mathbb{R}^{n \times n} \end{cases}$$

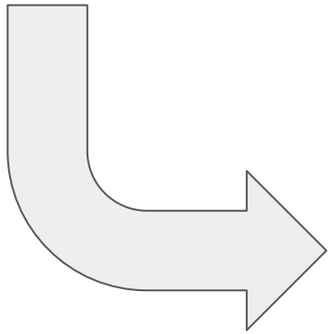
$$L^{(G)} = D - A$$

$$\lambda_1(L^{(G)}) \leq \lambda_2(L^{(G)}) \leq \cdots \leq \lambda_n(L^{(G)})$$

GIST (GRAPH STYLE AND CONTENT)

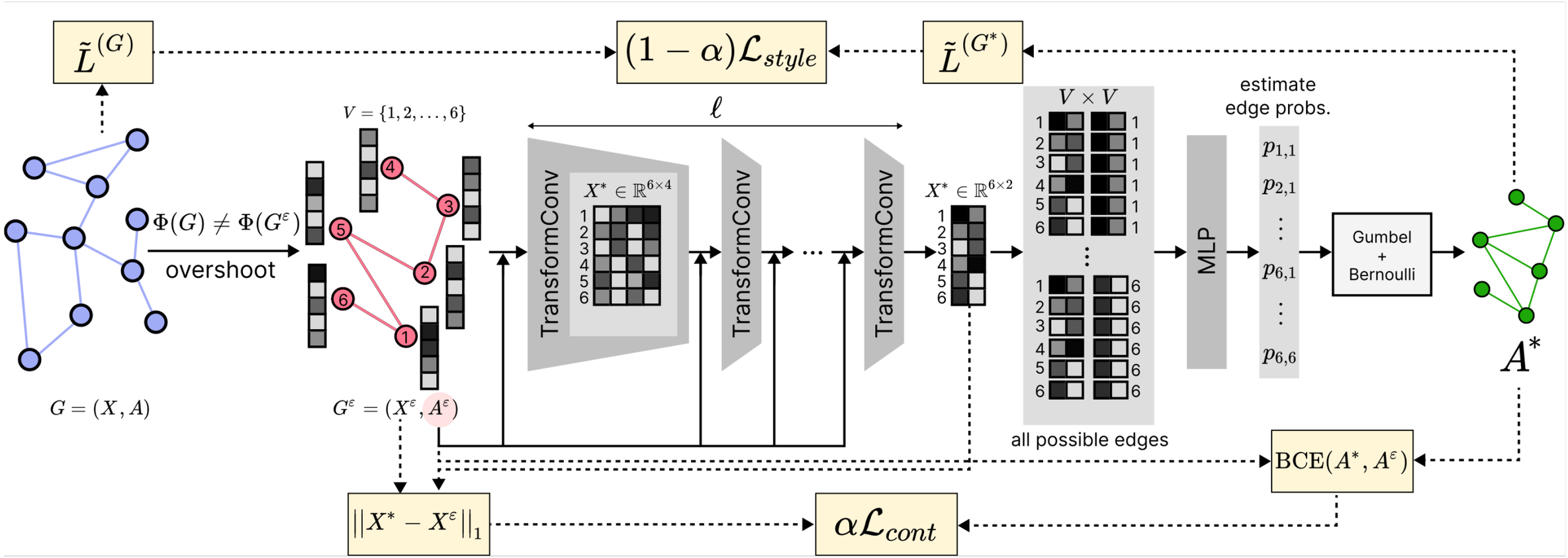
$$L^{(G)} = D - A$$

$$\lambda_1(L^{(G)}) \leq \lambda_2(L^{(G)}) \leq \cdots \leq \lambda_n(L^{(G)})$$



capture **global structural patterns**, e.g., connectivity and symmetry, that are largely **invariant** to specific node identities

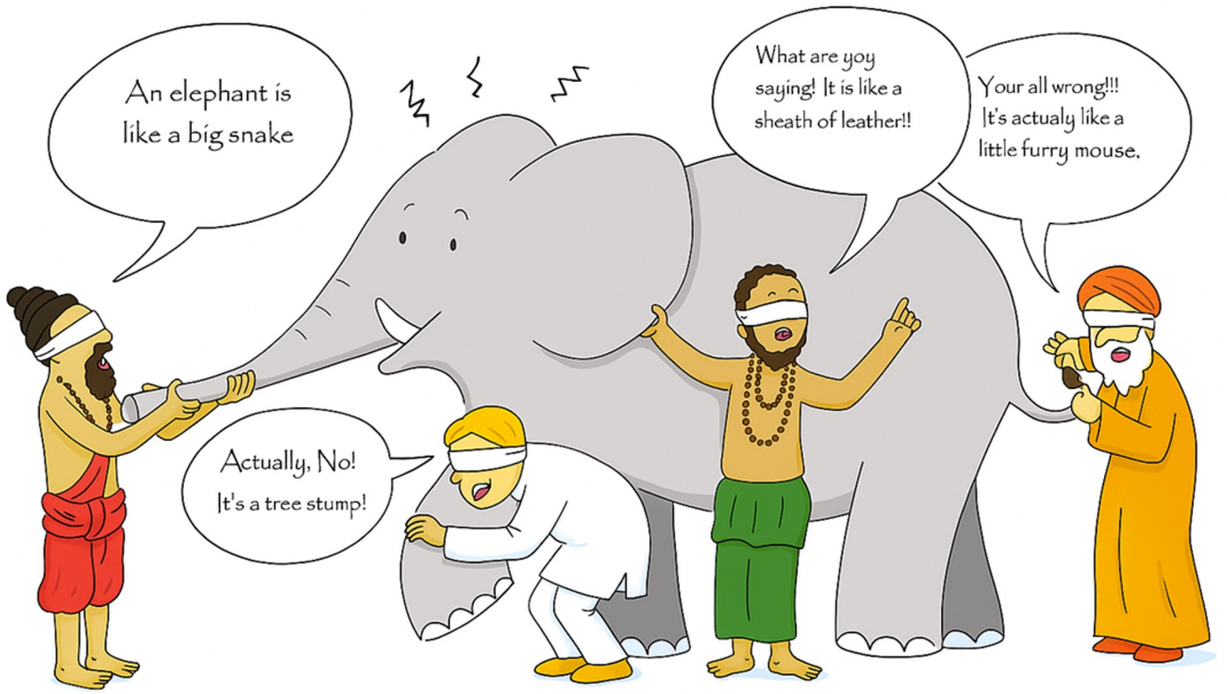
GIST (GRAPH STYLE AND CONTENT)



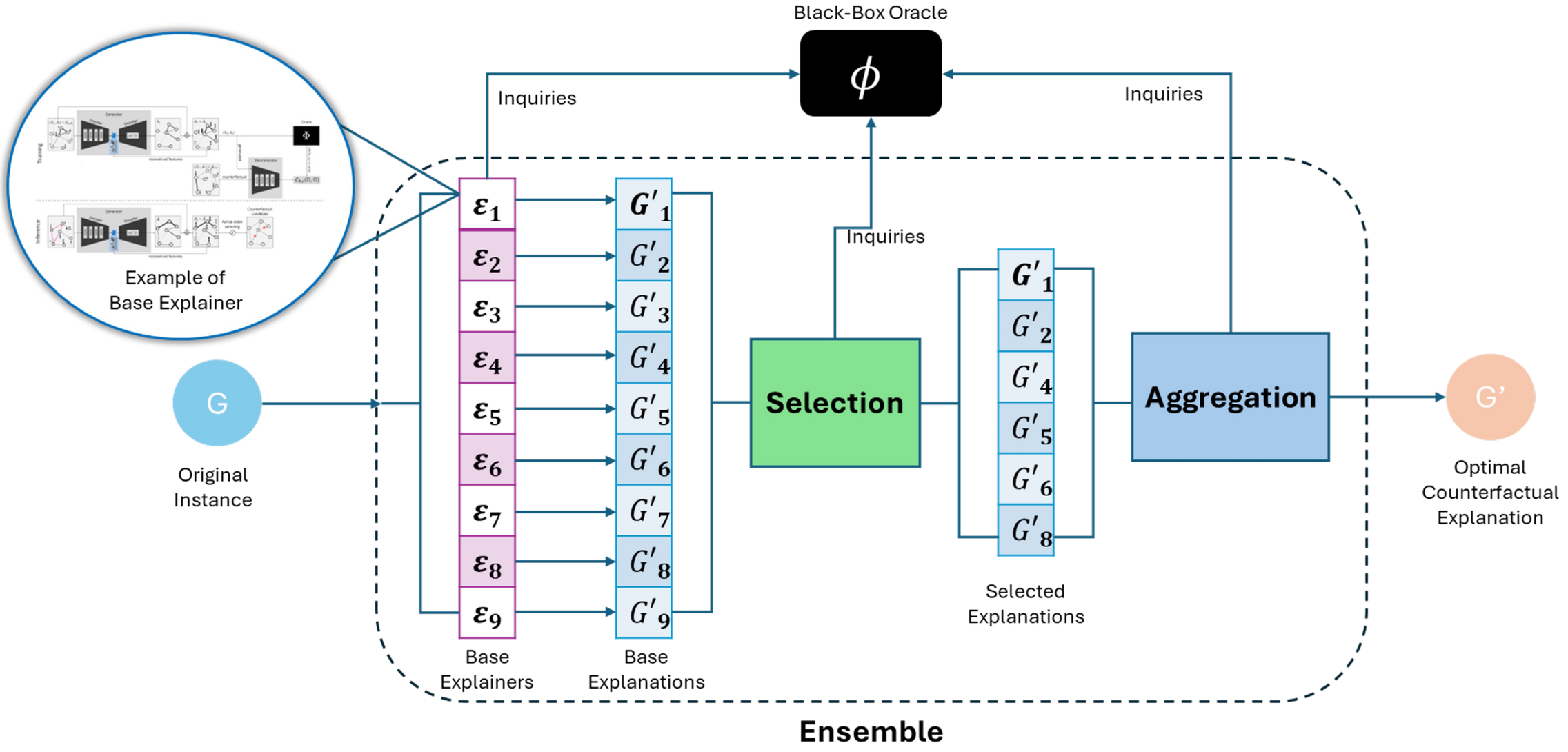
Meta-Explainers

Combining Explainers

Can we combine the strengths of different explainers into an ensemble that performs well across different domains?



Ensemble Strategy: Selection and Aggregation



Ideal-Point Multi-Criteria Selection (IPMCS)

Given an explainee instance G, an oracle Φ, and a set of base explainers $\mathbb{E} = \{\mathcal{E}_\Phi^1, \mathcal{E}_\Phi^2, \dots, \mathcal{E}_\Phi^K\}$ then IPMCS performs the following steps:

i. Get the set of explanations $\mathbb{C} = \bigcup_{i=1}^{K} \mathcal{E}_\Phi^i(\mathrm{G})$

ii. Given a set of criteria $H = \{h_1, h_2, \dots, h_k\}$ calculate an ideal point $Z = [z_1, z_2, \dots, z_k]$ where $z_i = \max\{h_i(G') \mid G' \in \mathbb{C}\}$

iii. Select the explanations $\mathrm{G}_\mathrm{i}' = \mathrm{argmax}_{\mathrm{G}_\mathrm{i}' \in \mathbb{C}} \mathrm{d}(\mathrm{Z}, \mathrm{H}(\mathrm{G}_\mathrm{i}'))$

Learning to Select Explainers

1) Dataset-Level Explainer Selection (DLES):

i. On training phase, uses the metrics set H and the ideal-point method on each training instance to determine the base explainer $\mathcal{E}^*_\Phi$ that performs the best on average.

ii. On inference time uses $\mathcal{E}^*_\Phi$ to explain every instance

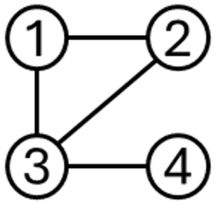
2) Instance-Level Explainer Selector (ILES):

i. On training phase, uses the explainer selected by IPMCS for each instance as the instance label creating a new training dataset for explainer selection, then trains a GNN on it.

ii. On inference time, uses the GNN to predict the best explainer for each instance based on the characteristics of that instance.

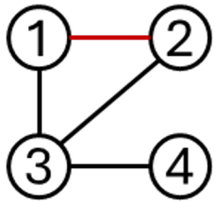
Explanation Aggregation: Change Frequency

Original Instance

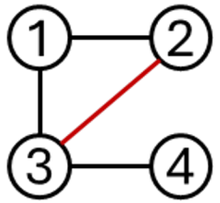


0	0	0	0
0	0	0	0
0	0	0	0
0	0	0	0

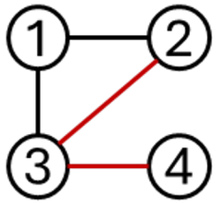
Counterfactual Instances



0	1	0	0
1	0	0	0
0	0	0	0
0	0	0	0



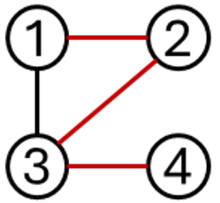
0	0	0	0
0	0	1	0
0	1	0	0
0	0	0	0



0	0	0	0
0	0	1	0
0	1	0	1
0	0	1	0

Changes Matrices

Aggregated changes



0	**.3**	0	0
.3	0	**.6**	0
0	**.6**	0	**.3**
0	0	**.3**	0

Change Frequency Matrix

Explanation Aggregation: Strategies

1) **Frequency Aggregation:** Applies to the original instance all the changes that appear with a frequency higher or equal than a value τ. Union ($\tau \geq 0$) and Intersection ($\tau \geq 1$) are special cases.

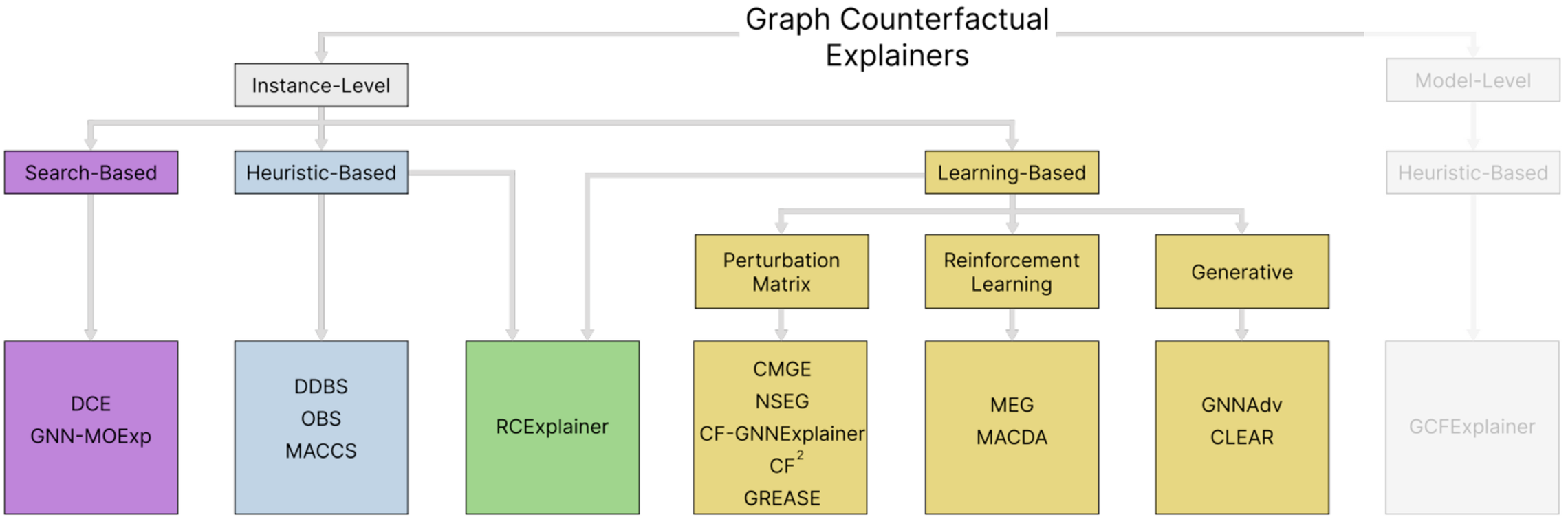
2) **Iterative Random Aggregation:** Iteratively selects random changes from the changes frequency matrix and applied them to the original instance until finds a counterfactual or reaches the maximum iteration threshold p.

3) **Stochastic Aggregation:** Consider the frequency of changes as probabilities and apply then, iteratively and in order, to the original instance until a counterfactual is found.

4) **Bidirectional Aggregation:** Behaves similarly to the Iterative Random in a first stage. In a second stage it tries to reduce the size of the solution by randomly undoing some of the changes while preserving the counterfactuality of the solution.

What's Next?

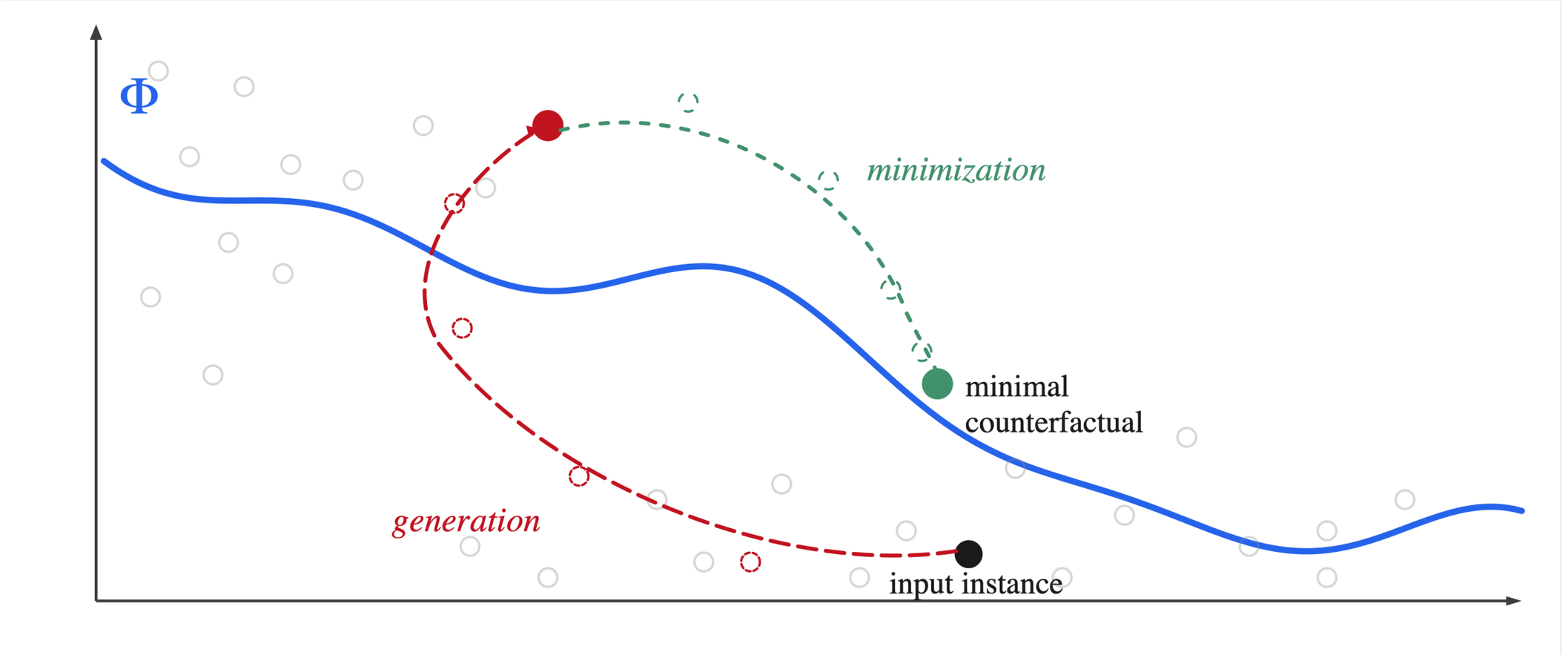
GCE Methods Taxonomy



On the Minimization of Graph Counterfactual Explanations

Rodrigo Garcia, Mario Alfonso Prado-Romero, Francesco Gullo, Giovanni Stilo

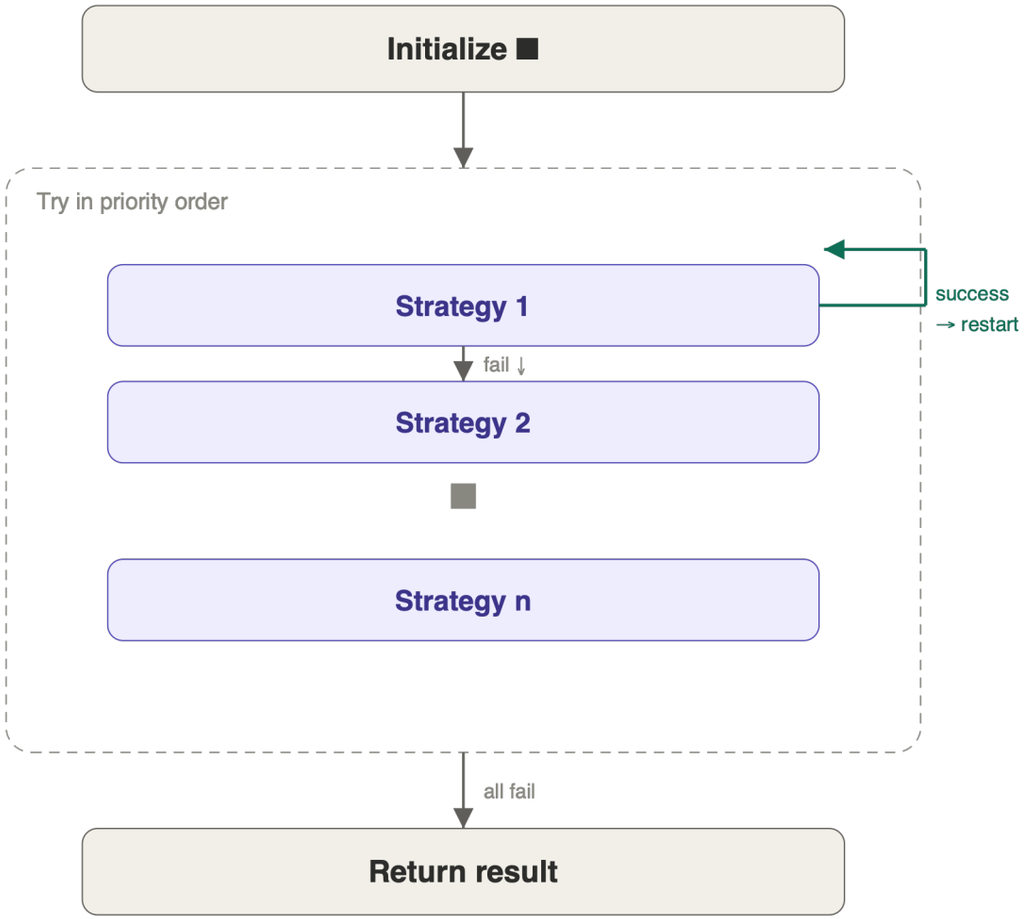
Generation-Minimization

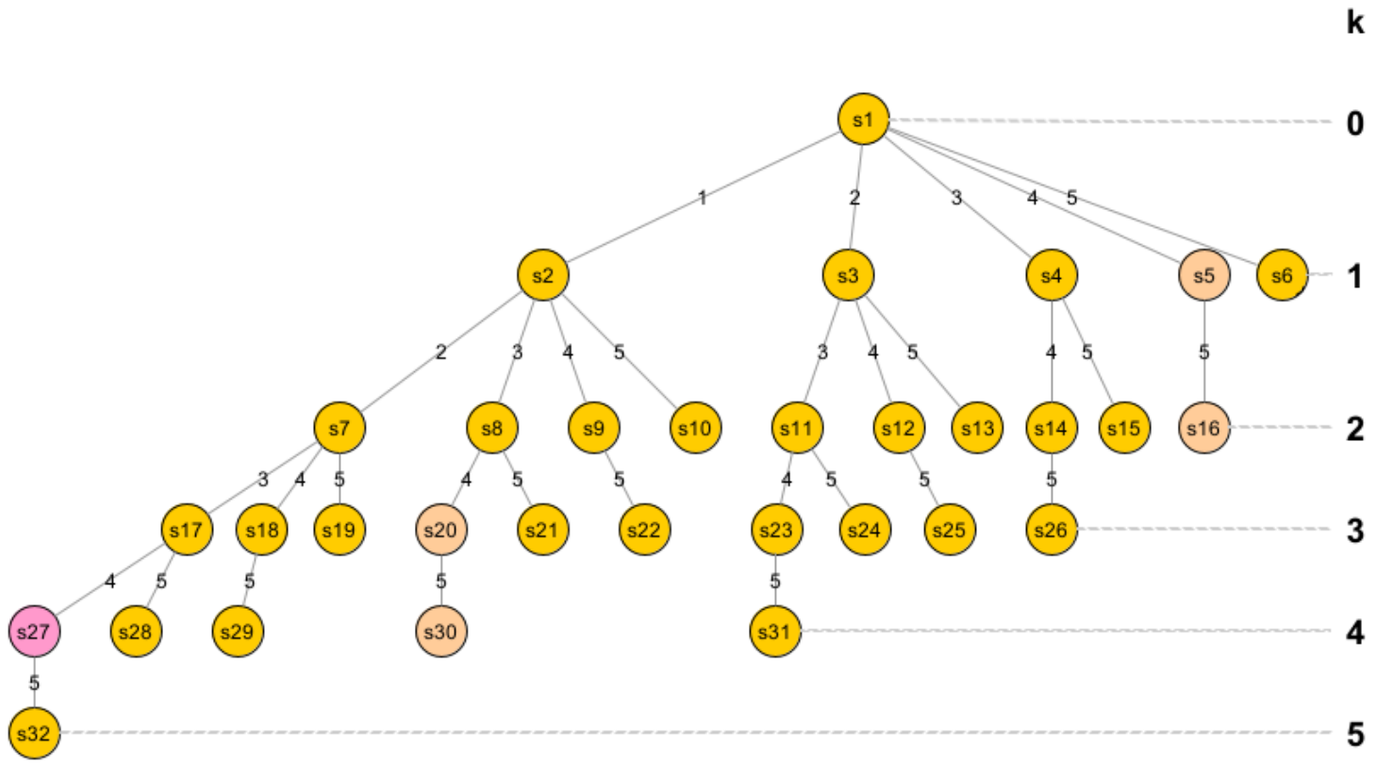


Problem 1 (GC-MIN). *Given a machine learning model (oracle)* $\Phi : \mathcal{G} \to \mathbb{R}$, *a graph* $G = (V, E, X) \in \mathcal{G}$, *and a GC* $G' = (V, E', X') \in \mathcal{G}$ *of* G, *modify* G' *so as to find* G'' *as*

$$G^* = \text{argmin}_{G'' \in \mathcal{G} | \Phi(G'') \neq \Phi(G)} |\Delta(G, G'')|. \quad (1)$$

Local Bounded Search





Human Readable Graph Counterfactual Explanation

Alejandra Montse Pena, Rodrigo Garcia, Giovanni Stilo

Related Works

Reference	Data Modality	Domain	Target Node	LLMs	Dataset	XIA
Fredes & Vitria[4]	Tabular	Any	×	GPT-4o	Adult Dataset	✓
Nguyen et al.[18]	Text	SA, NLI, HS	×	Llama2(7B/70B), Mistral(7B/56B), GPT3.5, GPT4	IMDb, SNLI	×
He et al.[8]	Graph	Molecule	×	GPT-3.5-Turbo, GPT-4	AIDS, Mutagenicity, BBBP, ClinTox, Tox21	×
Giorgi et al.[6]	Graph	Any	✓	Qwen2.5 (0.5B, 1.5B, 3B, 7B, 14B)	Cora, Citeseer	✓
Our proposal	Graph	Any	×	Llama (3B, 8B, 8B-Instruct) [currently]	TreeCycles [currently]	✓

Proposed Framework

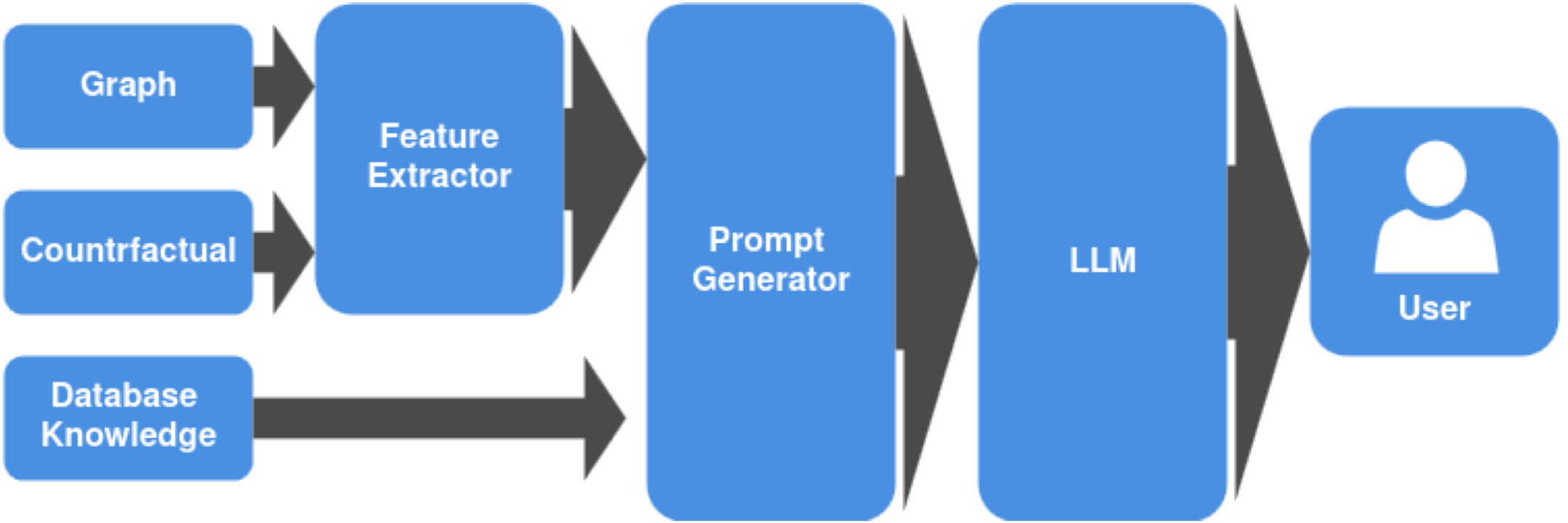


Figure 1: Illustration of the pipeline to explain counterfactual graphs with LLMs.

Open Research Questions

- **RQ1: Can we use text to represent a graph and then generate text counterfactuals and then go back to a graph repr?**
  - *This could maybe address the semanticity and actionability aspect of counterfactuals.*
- **RQ2:** How can we incorporate **domain knowledge** into the explanation methods?
- **RQ3**: **How uncertain** must the oracle be such that the produced **counterfactual is adversarial**?
- **RQ4**: When a method can be consider an **adversarial** attack one rather than **counterfactual explanation**?

Thanks for your attention!

QUESTIONS?

Workshop on Machine Unlearning

Co-Located ECML-PKDD

Naples 7-11 September

https://aiimlab.org/events.html

EDBT ICDT 2026 TAMPERE