FUJITSU

GNNXEMPLAR

EXEMPLARS TO EXPLANATIONS

THE FIRST NATURAL LANGUAGE GLOBAL EXPLAINER FOR GNNS

Burouj Armgaan, Eshan Jain, Harsh Pandey, Mahesh Chandran, Sayan Ranu

BLACKBOXES & EXPLAINERS

Black box Model whose reasoning is unavailable or too complex

Explainer Provides post-hoc reasoning behind black box predictions

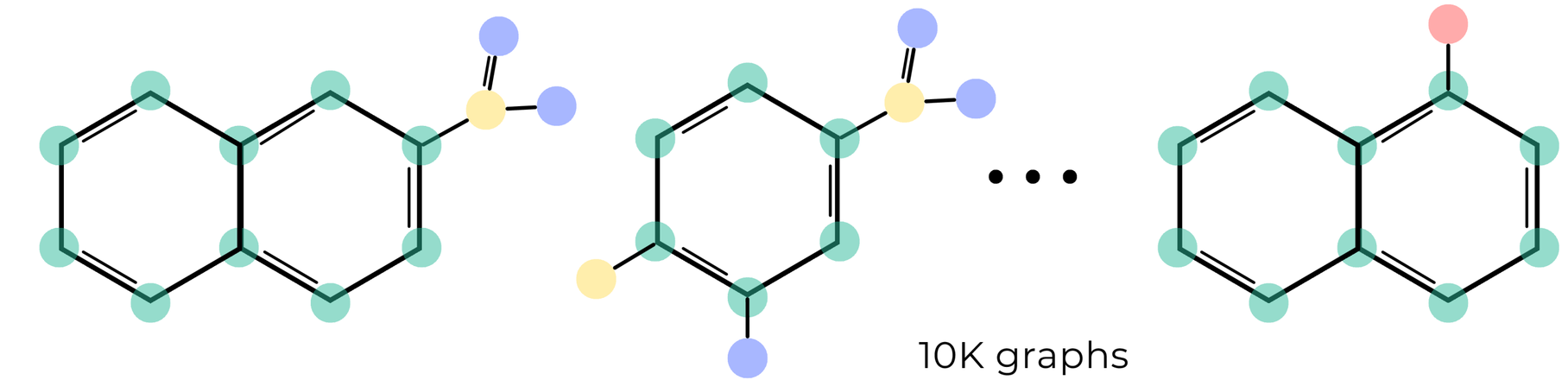
Three aspects

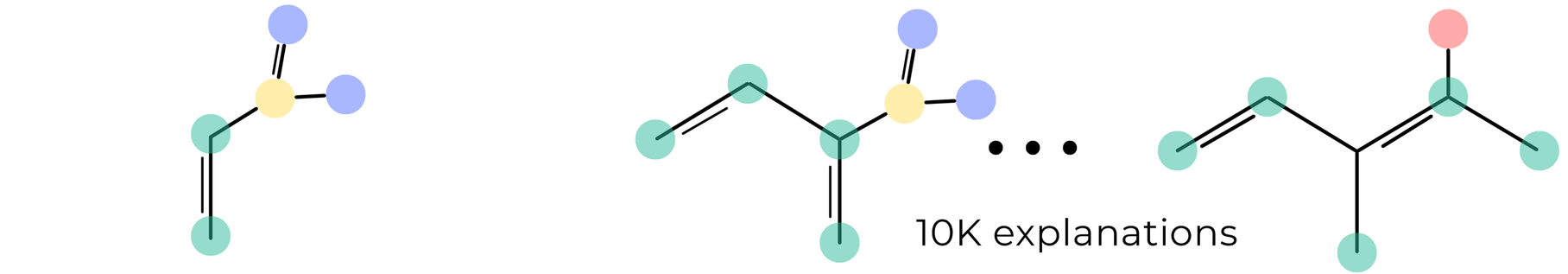
Validates correct reasoning

Highlights incorrect reasoning

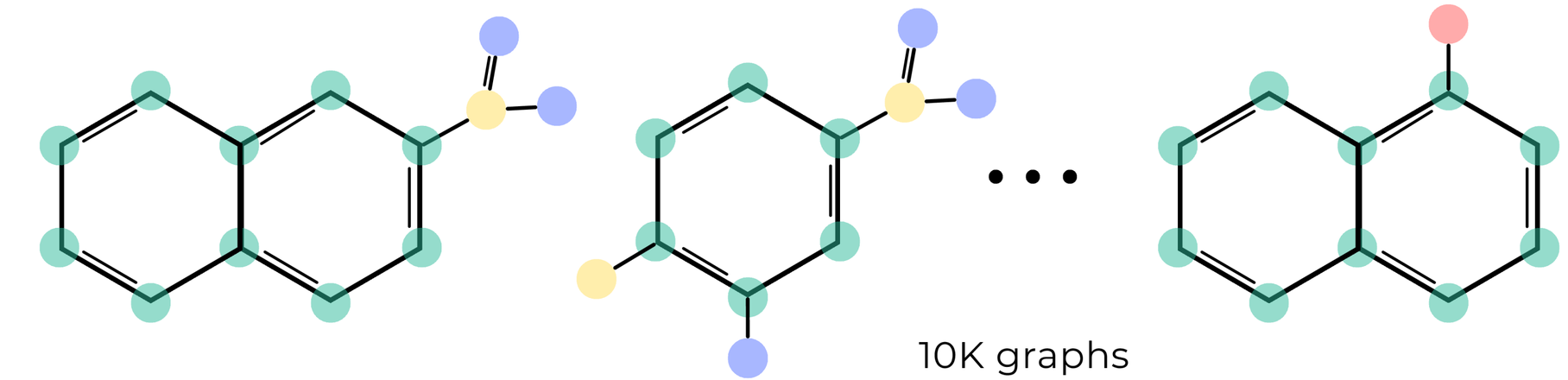
Insights when correct reasoning is unknown

LOCAL EXPLAINERS

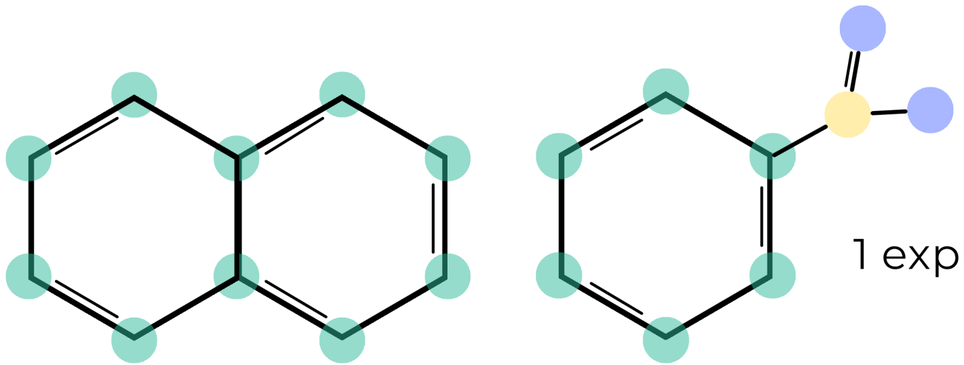




GLOBAL EXPLAINERS



10K graphs



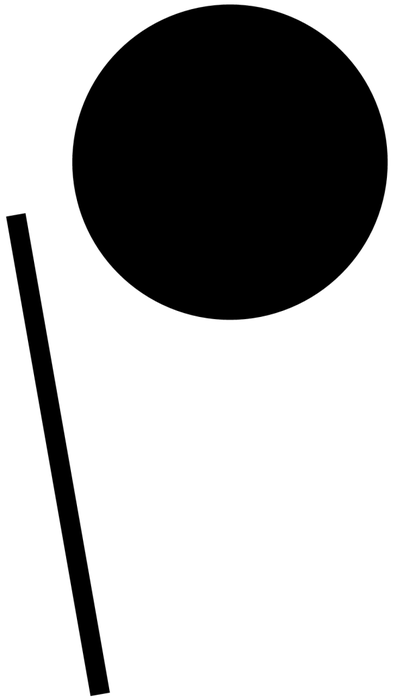
1 explanation

GLOBAL OVER LOCAL

ML models **don’t learn** rules for **individual** nodes/graphs
They learn rules for **entire classes**

Human interpretability: **1 vs 1000** explanations

Local explainers require **manual labour**
for class-level understanding



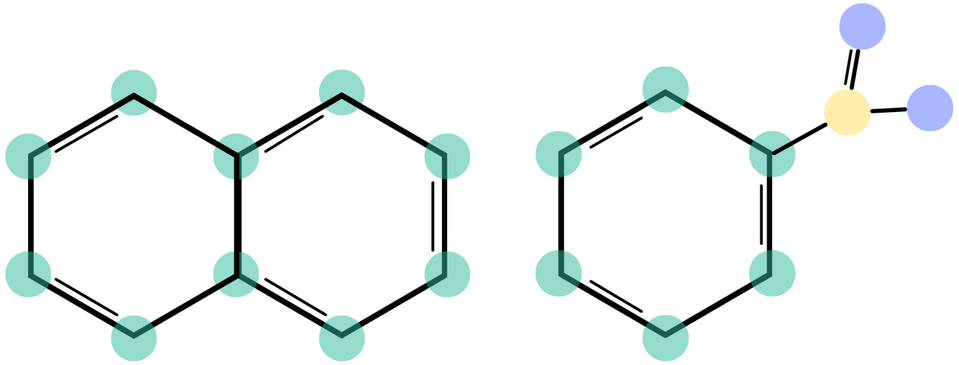
LIMITATIONS OF EXISTING GLOBAL EXPLAINERS

RECURRING STRUCTURES

1

Assume classes can be captured through **recurring subgraphs** or motifs

Reasonable in **molecules**



But, what about **complex networks?**

- Citation networks
- Social networks
- Financial transactions

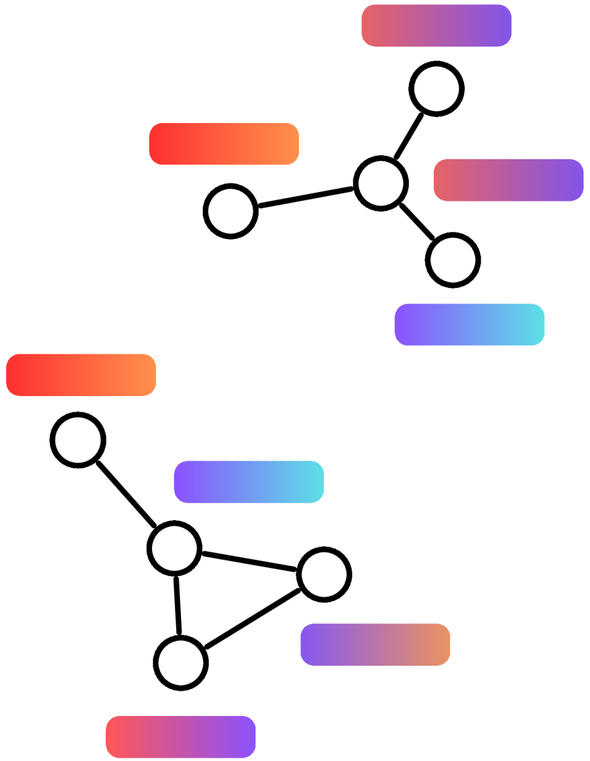
CATEGORICAL FEATURES

2

The reliance on motifs
forces the use of **isomorphism**

Isomorphism **fails on rich, continuous features**
Pretty common in graphs

Isomorphism how?
No recurring structure. Rich feature vectors



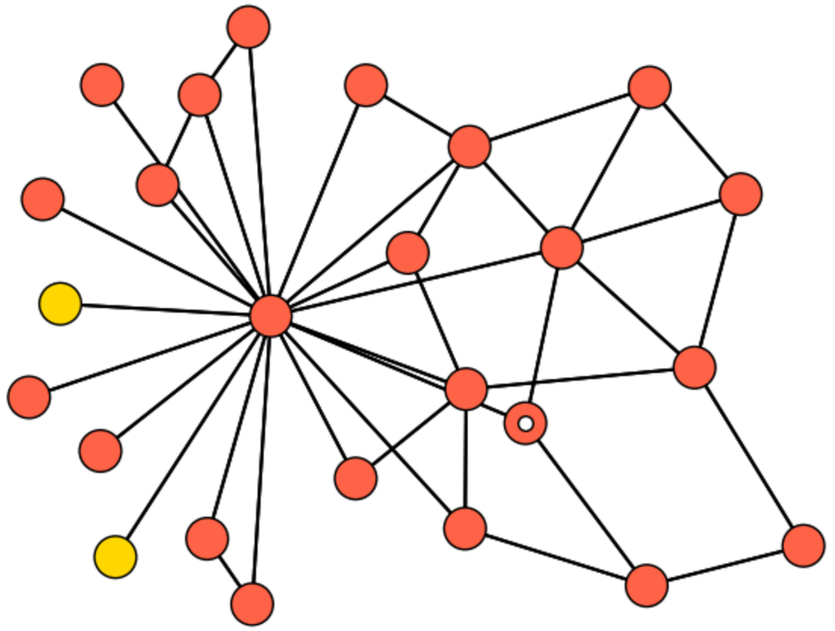
OPEN TO INTERPRETATION

3

A subgraph is **open** to interpretation

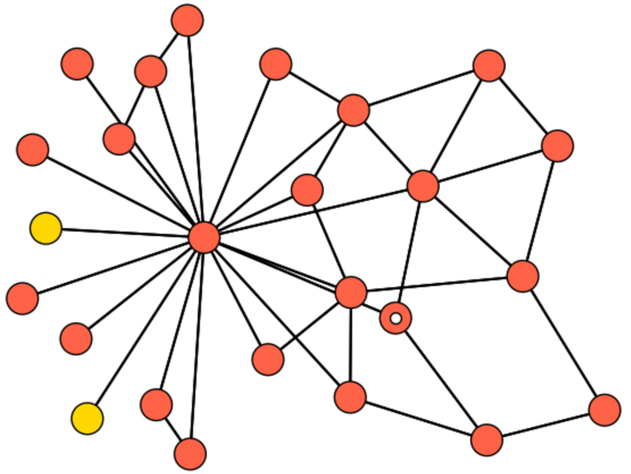
Users may draw **different interpretations** from the **same subgraph**, unaware of the explainer's intended meaning.

Graph properties like degree, clustering, homophily are **difficult to convey precisely** via subgraphs.



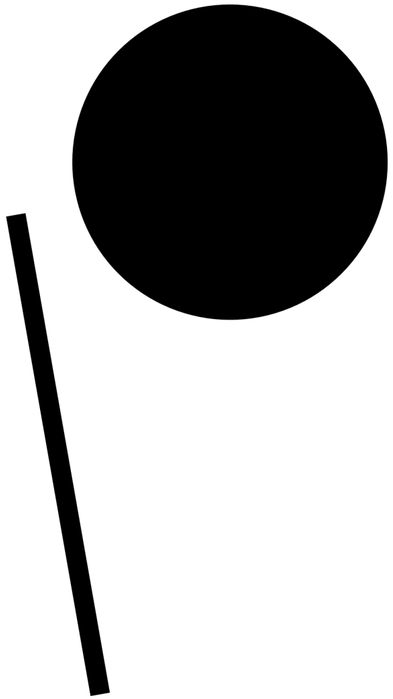
EXAMPLE

Nodes Publications
Edges Citations
Features Abstract as Bag-of-Words
Classes ML, Theory, Probabilistic methods, ...



CLASS: PROBABILISTIC METHODS

Probabilistic Methods papers include **keywords** like “Bayesian network,” “Markov chain,” “density estimation,” “Gibbs sampling,” or “MCMC,” and/or at least **20–80% of their 1-hop and 2-hop neighbors** are also Probabilistic Methods publications.



HOW?
EXEMPLAR THEORY

EXEMPLAR

Cambridge “A typical or good example of something”

These are **typical** members of a **population**

They **exemplify** the **signature** characteristics of their **population**

EXEMPLAR THEORY

Rooted in cognitive psychology

Posits that humans assign categories by comparing new stimuli with previously encountered instances, called **exemplars**

AN EXEMPLAR & ITS SIGNATURE

A **node** that **typifies** the topology and features of others in **its predicted class**

A **diverse** population can have **multiple** exemplars

The **shared distinguishing** traits of the exemplar's **population**

Represented as a **boolean function** composed of interpretable conditions

AN EXEMPLAR & ITS SIGNATURE

Probabilistic Methods
Features include keywords like “Bayesian network”, “density estimation”, and/or 60% of 1-2 hop neighbors are of the same category.

This **description** is the signature

The **node that best expresses** these traits is its exemplar

PROBLEM FORMULATION

Graph	$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$
Features	$\mathbf{X} = \{\mathbf{x}_v \mid v \in \mathcal{V}\},\ \mathbf{x}_v \in \mathbb{R}^d$
Labels	$\forall v \in \mathcal{V},\ Y_v \in \{y_1, ..., y_c\}$
GNN	$\forall v \in \mathcal{V},\ \Phi(v) \in \{y_1, ..., y_c\}$

PROBLEM FORMULATION

Identify the **exemplars** for each class $\mathcal{E}_i \in \{e_1, e_2, ..., e_b\}$

Extract their **signatures** as boolean python functions $\sigma_e(v)$

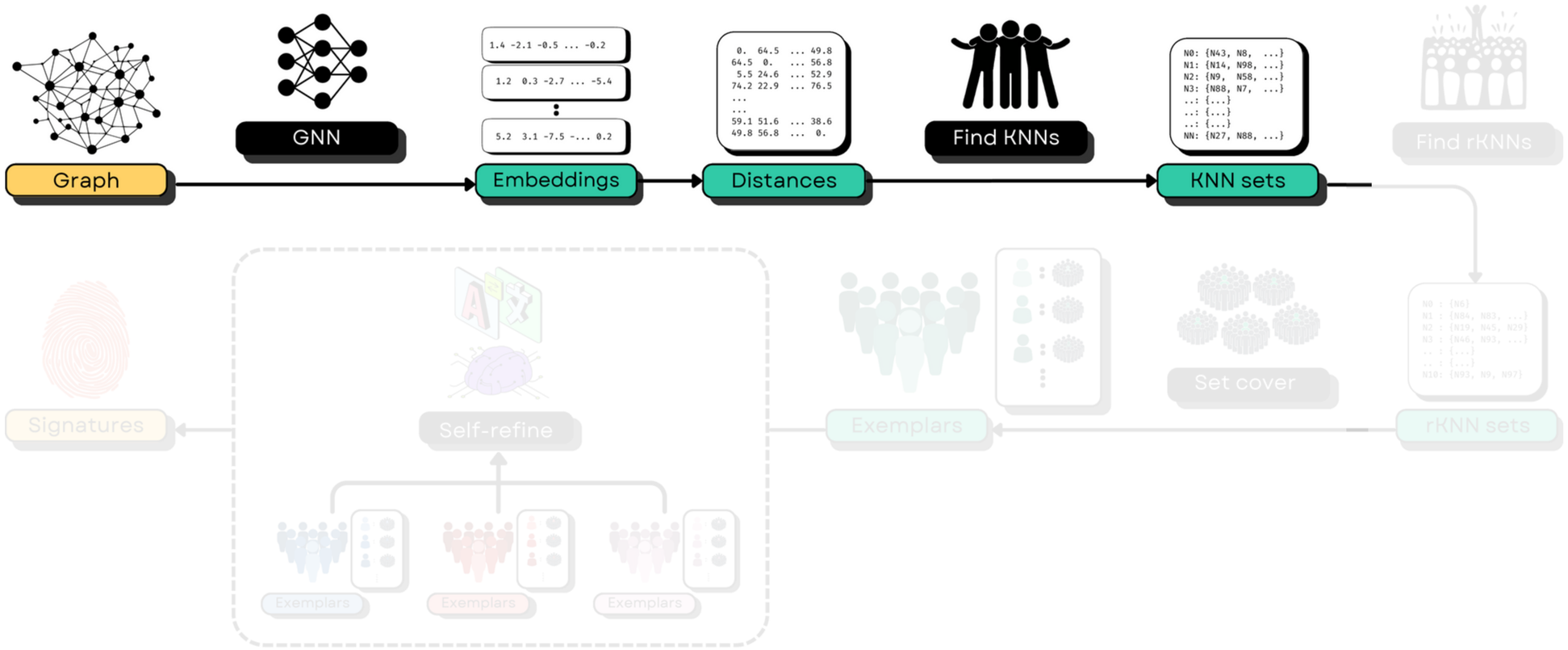
Combine logically to form a **class signature** $f_i(v) = \bigvee_{e \in \mathcal{E}_i} \sigma_e(v)$

Present the same in **natural language** $f_i(v)$ to text

KNN IN THE EMBEDDING SPACE

1

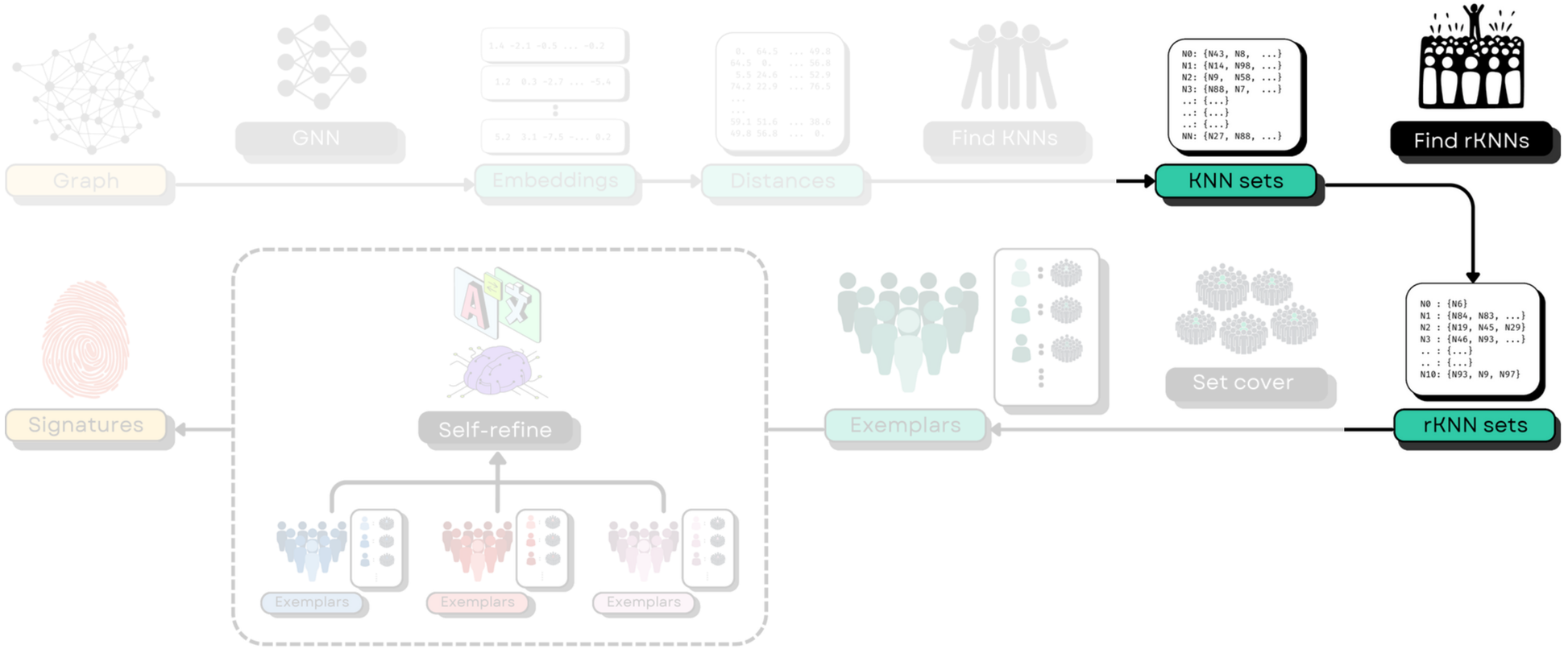
Compute k-nearest neighbors in the GNN **embedding space**
Distances in the embedding space represent **functional similarity**



REVERSE KNN

2

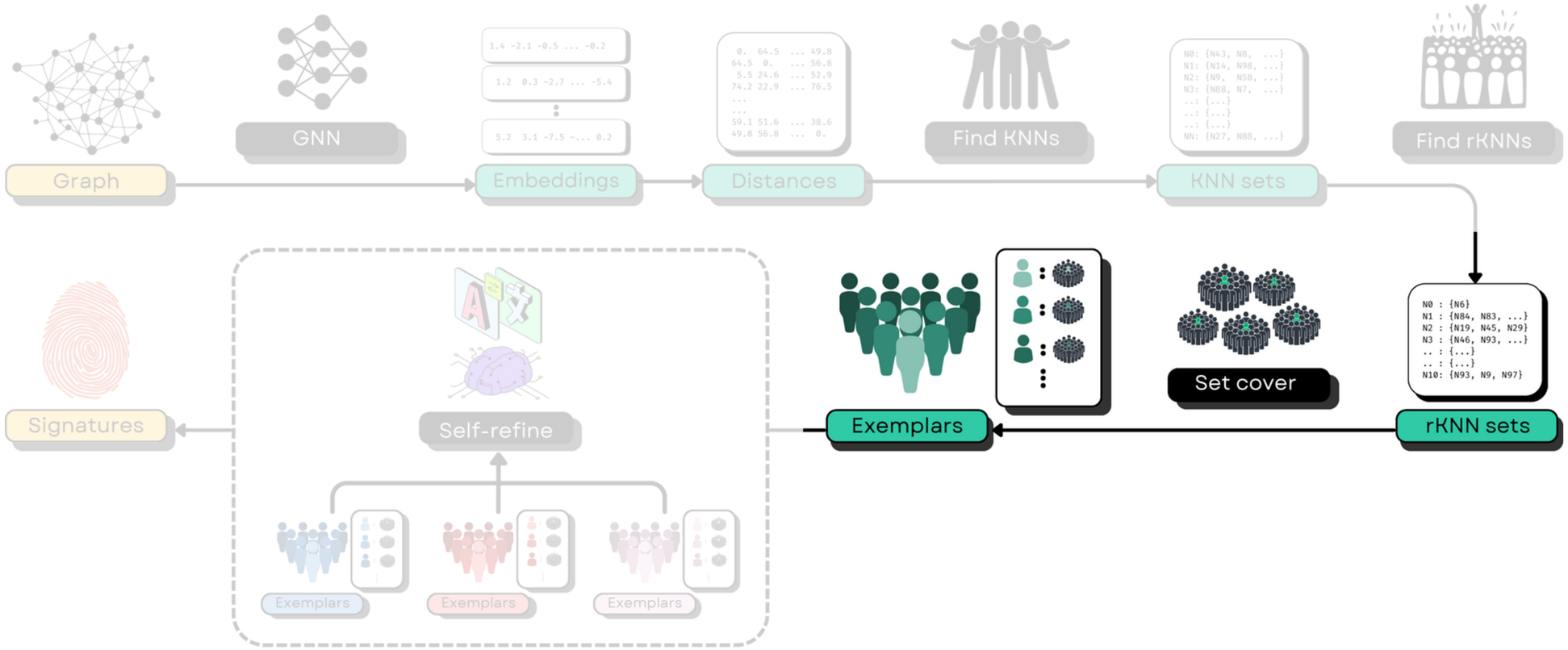
Identify nodes that exists in the K-nearest neighbor of lots of nodes.
Such nodes are **popular** nodes, a **representative** nodes, **exemplars.**



COVERAGE MAXIMIZATION

3

Exemplars may have **overlapping populations.** Pick the ones that represent the class broadly and without redundancy.



PROBLEM FORMULATION

Identify the **exemplars** for each class $\mathcal{E}_i \in \{e_1, e_2, ..., e_b\}$

Extract their **signatures** as boolean python functions $\sigma_e(v)$

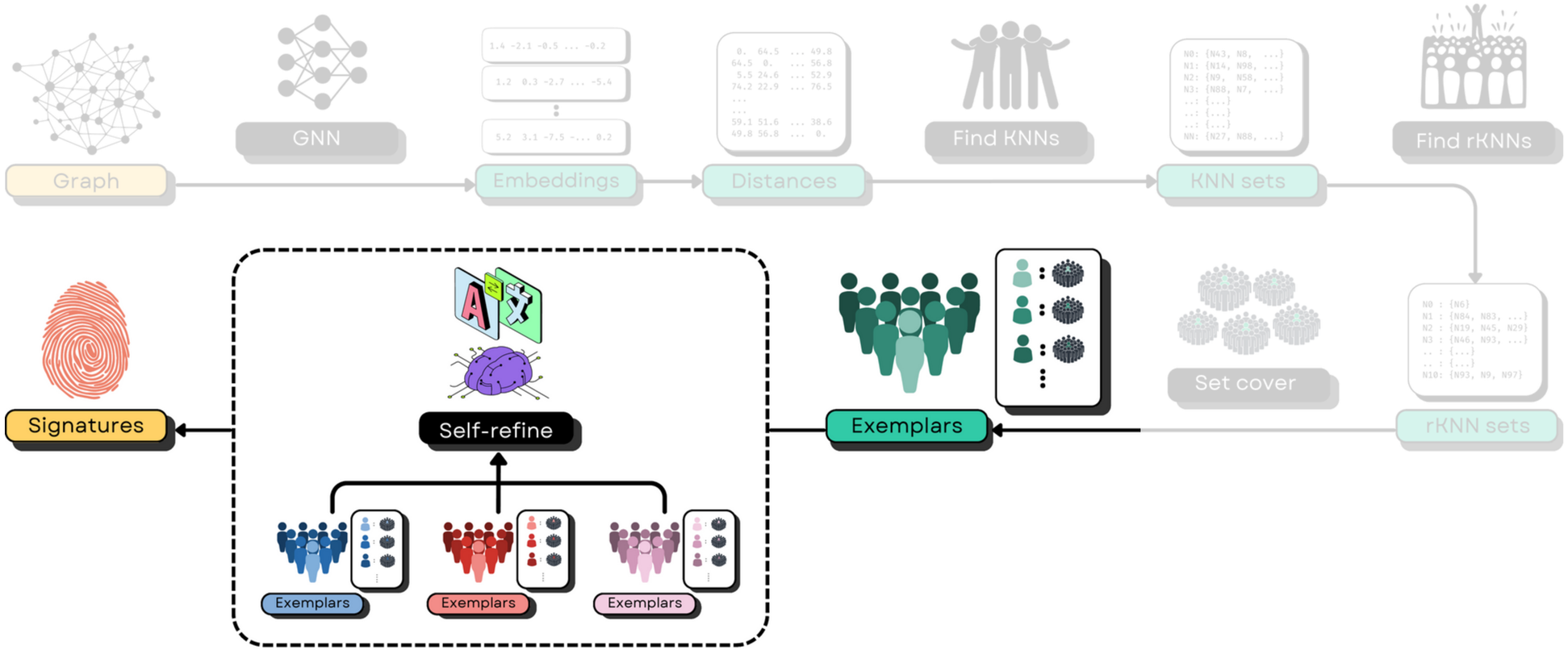
Combine logically to form a **class signature** $f_i(v) = \bigvee_{e \in \mathcal{E}_i} \sigma_e(v)$

Present the same in **natural language** $f_i(v)$ to text

THE HARD PART

4

How do we make an LLM find the signature **across graphs?**
LLMs are not designed for graph reasoning



SELF-REFINE

4

We do not always generate our best output on our **first try.**

Self-refine is an iterative process of creating an **initial draft** and subsequently **refining it** based on **self-provided feedback**

A fundamental characteristic of human problem-solving

- Drafting an email
- Writing code
- Designing an algorithm
- Writing a research paper

PROMPT

Describe the task

Specify the output skeleton

Provide exemplar and training nodes

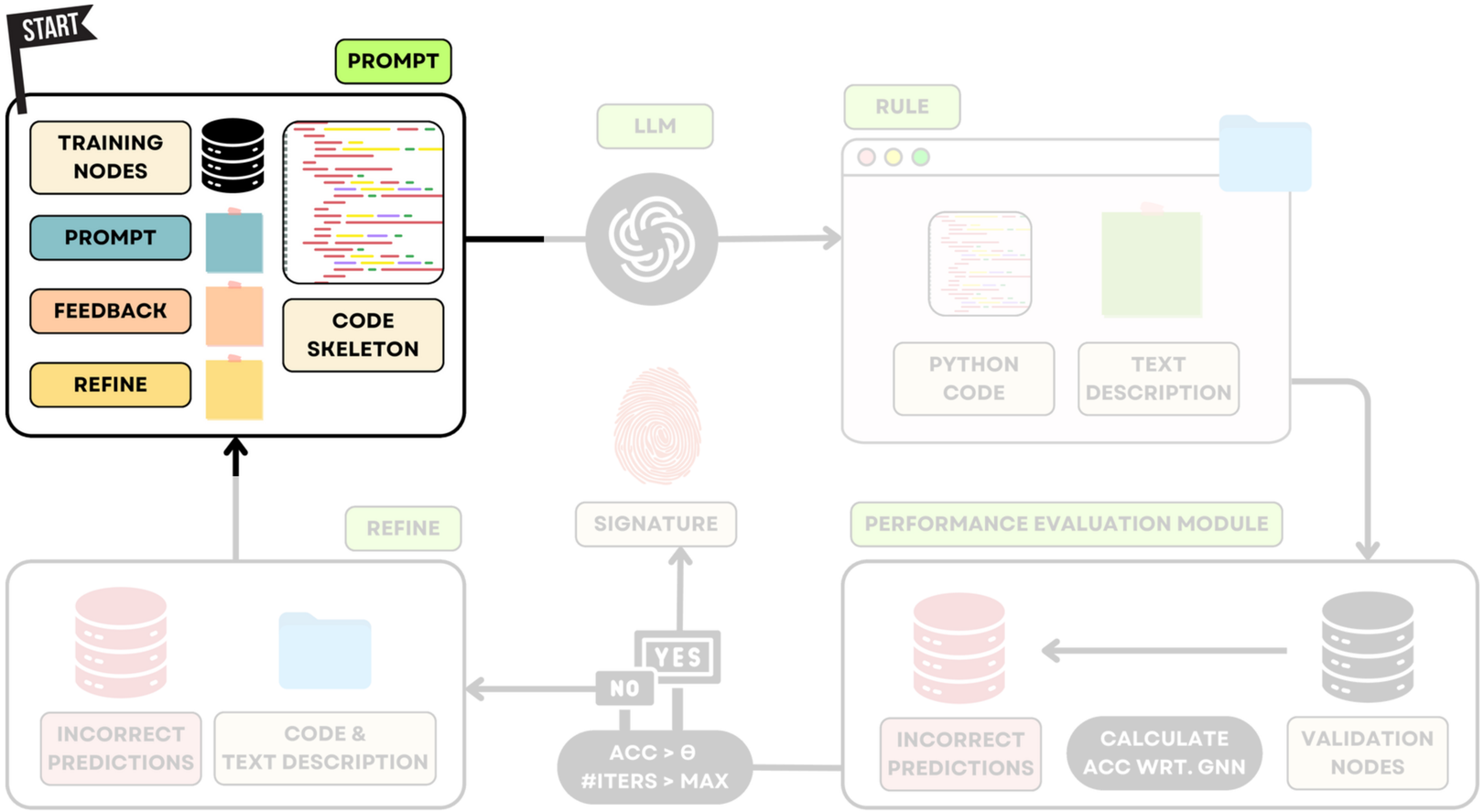
Success hinges on

which neighborhood info is passed and how

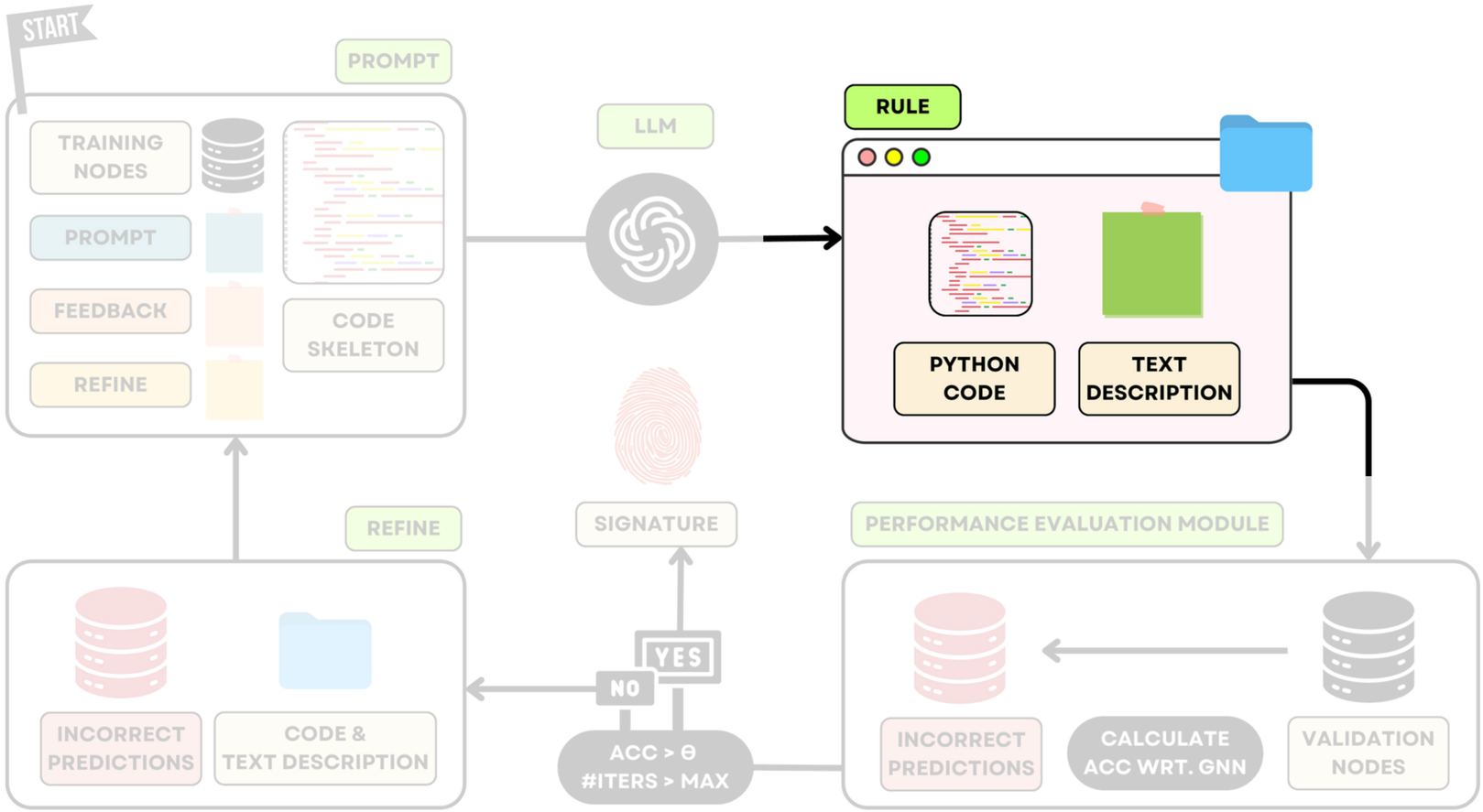
More on this in our poster session

Madaan, Aman, et al. "Self-refine: Iterative refinement with self-feedback." NeurIPS 2023

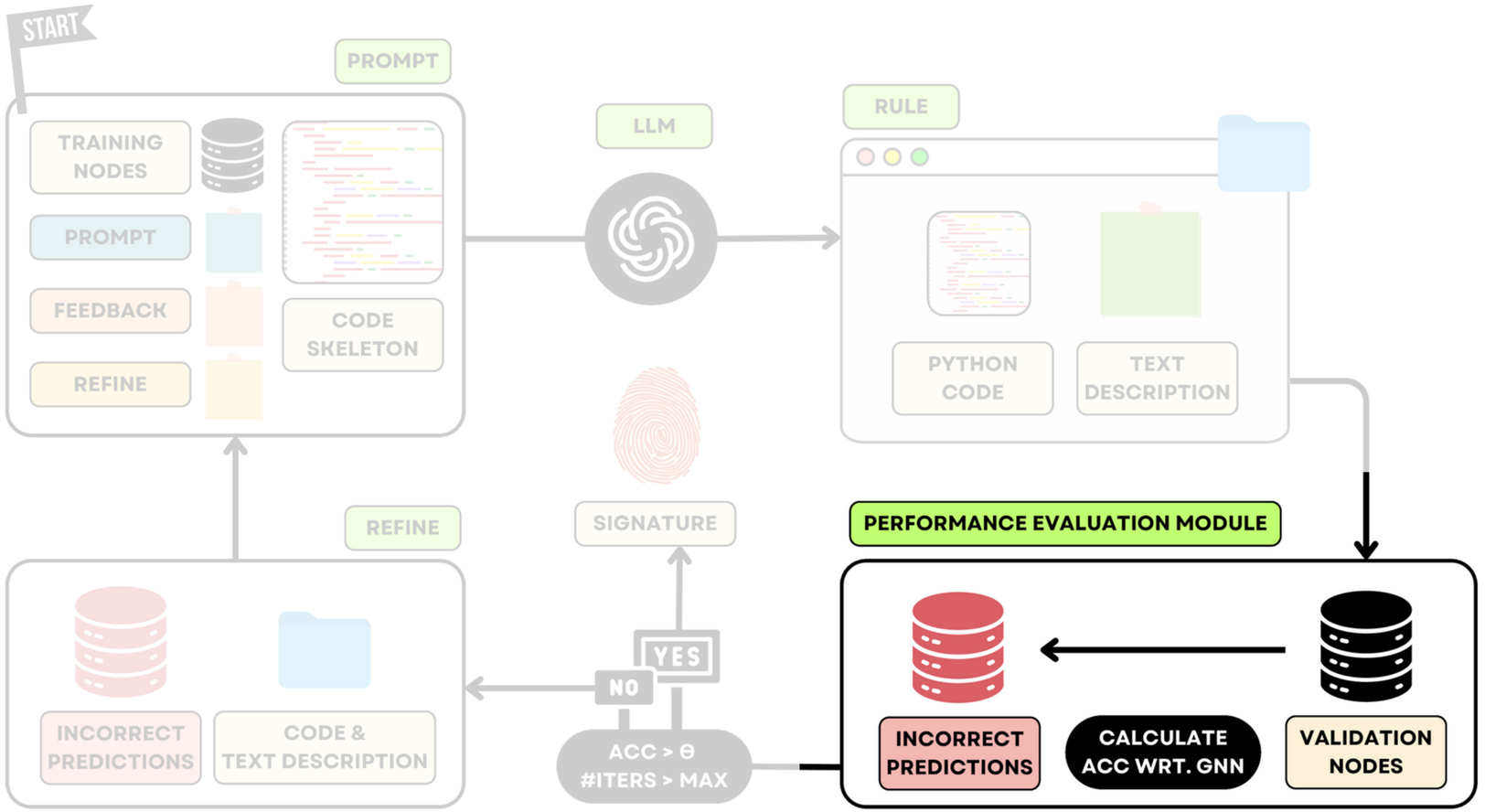
SELF-REFINE


START
PROMPT
TRAINING NODES
PROMPT
FEEDBACK
REFINE
CODE SKELETON
LLM
RULE
PYTHON CODE
TEXT DESCRIPTION
SIGNATURE
PERFORMANCE EVALUATION MODULE
REFINE
INCORRECT PREDICTIONS
CODE & TEXT DESCRIPTION
YES
NO
ACC > θ
#ITERS > MAX
INCORRECT PREDICTIONS
CALCULATE ACC WRT. GNN
VALIDATION NODES

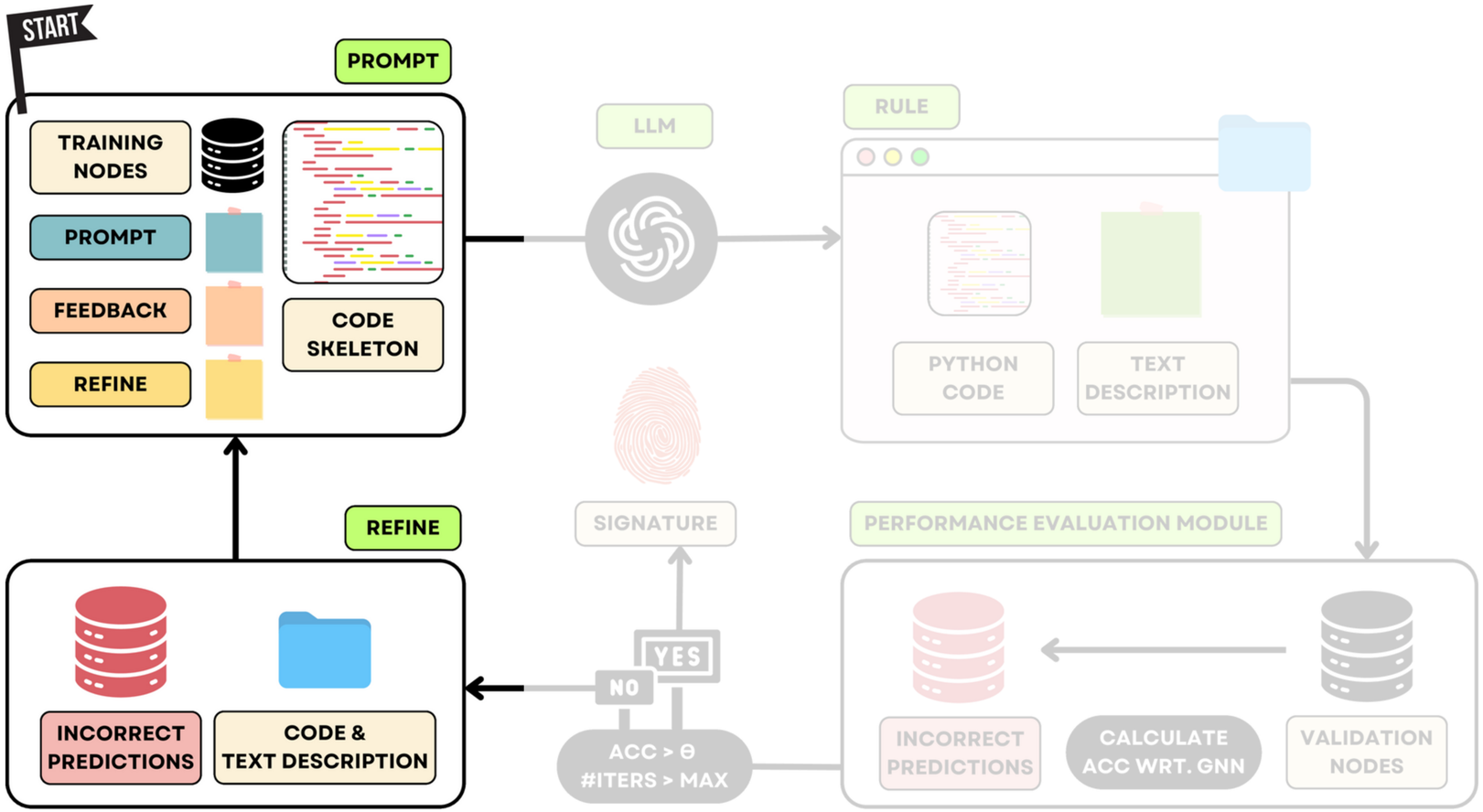
SELF-REFINE


START
PROMPT
TRAINING NODES
PROMPT
FEEDBACK
REFINE
CODE SKELETON
LLM
RULE
PYTHON CODE
TEXT DESCRIPTION
SIGNATURE
PERFORMANCE EVALUATION MODULE
REFINE
YES
NO
ACC > θ
#ITERS > MAX
INCORRECT PREDICTIONS
CODE & TEXT DESCRIPTION
INCORRECT PREDICTIONS
CALCULATE ACC WRT. GNN
VALIDATION NODES

SELF-REFINE


START
PROMPT
TRAINING NODES
PROMPT
FEEDBACK
REFINE
CODE SKELETON
LLM
RULE
PYTHON CODE
TEXT DESCRIPTION
PERFORMANCE EVALUATION MODULE
INCORRECT PREDICTIONS
CALCULATE ACC WRT. GNN
VALIDATION NODES
SIGNATURE
YES
NO
ACC > θ
#ITERS > MAX
REFINE
INCORRECT PREDICTIONS
CODE & TEXT DESCRIPTION

SELF-REFINE


START
PROMPT
TRAINING NODES
PROMPT
FEEDBACK
REFINE
CODE SKELETON
LLM
RULE
PYTHON CODE
TEXT DESCRIPTION
SIGNATURE
PERFORMANCE EVALUATION MODULE
REFINE
INCORRECT PREDICTIONS
CODE & TEXT DESCRIPTION
YES
NO
ACC > θ
#ITERS > MAX
INCORRECT PREDICTIONS
CALCULATE ACC WRT. GNN
VALIDATION NODES

FEEDBACK

Show current signature

Report evaluation

Highlight errors

Detail mistakes and Analyze causes

Request specific refinement

REFINE

Show full iteration history

Summarize actionable steps based on feedback

State revision task for a better signature

PROBLEM FORMULATION

- Identify the **exemplars** for each class $\mathcal{E}_i \in \{e_1, e_2, ..., e_b\}$
- Extract their **signatures** as boolean python functions $\sigma_e(v)$
- Combine logically to form a **class signature** $f_i(v) = \bigvee_{e \in \mathcal{E}_i} \sigma_e(v)$
- Present the same in **natural language** $f_i(v)$ to text

EXAMPLE

```python
def classify_class_0(node_description):
    features = node_description.get('features', '')
    one_hop  = node_description.get('1-hop',  {}).get('
       neighbor_class_freq', {})
    two_hop  = node_description.get('2-hop',  {}).get('
       neighbor_class_freq', {})

    # Exemplar #1 signature
    score1  = 0.5 * int(any(k in features
                        for k in ['ILP','meta-knowledge','
                            hypothesis space']))
    score1 += 0.3 * int(one_hop.get(0,0) > 0.8)
    score1 += 0.2 * int(two_hop.get(0,0) > 0.8)
    cond1   = (score1 >= 0.5)

    # Exemplar #2 signature
    score2  = 0.5 * int(any(kw in features.lower()
                        for kw in ['logic program',
                                   'inductive logic programming',
                                   'ilp']))
    score2 += 0.3 * int(one_hop.get(0,0) > 0.7)
    score2 += 0.2 * int(two_hop.get(0,0) > 0.5)
    cond2   = (score2 >= 0.5)

    return cond1 or cond2
```

2 exemplars

Their signatures

Class signature

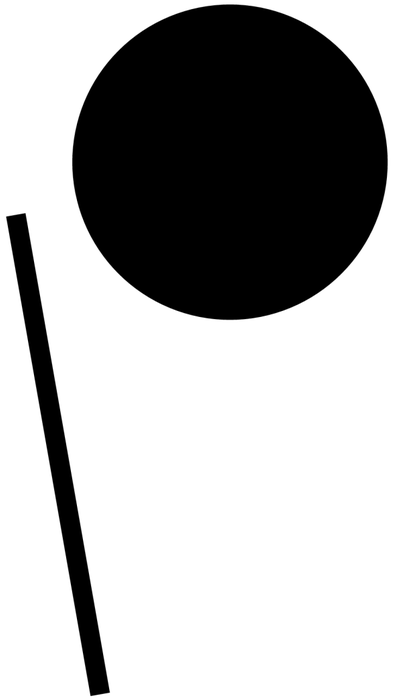
Applicable

Interpretable

Class 0 (Rule Learning)

A publication belongs to the *Rule Learning* class if its text mentions “inductive logic programming,” “meta-knowledge,” or “hypothesis space,” *or* if a large majority of its 1-hop and 2-hop citation neighbors also belong to Rule Learning.

Translated to text



RESULTS

A GOOD GNN EXPLAINER

Doesn’t assume recurring structures

Extends to rich features

Faithful

Scales well

Explicit meaning

Human preference

QUANTITATIVE

Baselines

Inapplicable (NA) Memory (OOM)

Timeouts (NF) Poor fidelity

GnnXemplar

Generalizes Better fidelity

Scales well

	Homophilous				Heterophilous			
	TAGCora	**Citeseer**	**WikiCS**	**arxiv**	**Amazon-R**	**Questions**	**Minesweeper**	**BA-Shapes**
GNNInterpreter	NA	0.50 ± 0.0	NA	NA	NA	NA	0.50 ± 0.0	0.47 ± 0.0
GCNeuron	0.51 ± 0.0	0.50 ± 0.0	OOM	OOM	0.56 ± 0.0	OOM	0.54 ± 0.0	0.50 ± 0.0
GLGExplainer	NF	NF	OOM	OOM	NF	OOM	0.22 ± 0.07	0.30 ± 0.09
GnnXemplar	$\mathbf{0.83 \pm 0.01}$	$\mathbf{0.92 \pm 0.03}$	$\mathbf{0.78 \pm 0.01}$	$\mathbf{0.84 \pm 0.01}$	$\mathbf{0.82 \pm 0.01}$	$\mathbf{0.92 \pm 0.01}$	$\mathbf{0.86 \pm 0.02}$	$\mathbf{0.93 \pm 0.00}$

No class-specific motifs

Rich features

A GOOD GNN EXPLAINER

- Doesn’t assume recurring structures
- Extends to rich features
- Faithful
- Scales well
- Explicit meaning
- Human preference

A HUMAN PREROGATIVE

Complex explanations **fail their purpose** even when accurate

Need to **judge** whether an explanation is understandable

Interpret | Merriam-webster
“to conceive in the light of individual belief, judgment, or circumstance”

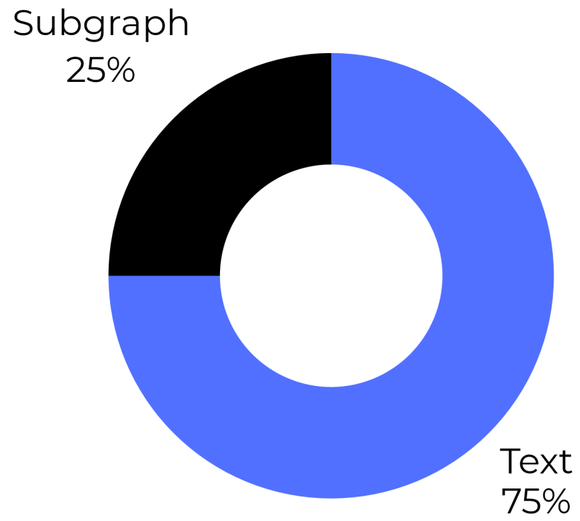
Because interpretation depends on individual thought,
only human assessment can measure it.

SURVEY

60 participants

5 A/B tests

300 total comparisons



Aggregate binomial	Text over subgraphs **overall?**
Per question binomial	Text over subgraph based on certain **dataset characteristics?**
McNemar's	Does an **individual consistency** pick text over subgraphs?

A GOOD GNN EXPLAINER

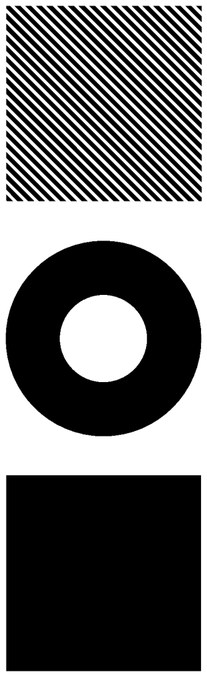
- Doesn’t assume recurring structures
- Extends to rich features
- Faithful
- Scales well
- Explicit meaning
- Human preference

GNNXEMPLAR

First GraphXAI technique to provide textual explanations & address these limitations

- Doesn’t assume recurring structures
- Extends to rich features
- Faithful
- Scales well
- Explicit meaning
- Human preference

Marks a shift in Graph XAI from subgraphs to natural language



THANK YOU

Have a great day

PAPER

CODEBASE

POSTER TODAY 4:30 — 7:30 PM
EXHIBIT HALL C,D,E #3801