PL4XGL: A Programming Language Approach to Explainable Graph Learning

Minseok Jeon, Jihyeok Park, and Hakjoo Oh

KOREA UNIVERSITY

PLDI 2024 @ Copenhagen, Denmark

Graph Machine Learning

Graph data → Machine Learning Model → Classification result

Graph data:

- Molecule
- Trading activity
- Health care
- Program
- …

Classification result:

- Postive or negative
- Fraud detection
- Safe or not
- Buggy or not
- …

Graph Machine Learning

- Mainstream: Graph Neural Network (unexplainable AI)

Graph data → Graph Neural Network → Classification result

Graph data:
- Molecule
- Trading activity
- Health care
- Program
- …

Classification result:
- Postive or negative
- Fraud detection
- Safe or not
- Buggy or not
- …

Graph Machine Learning

- Mainstream: Graph Neural Network (unexplainable AI)

Graph data → Graph Neural Network → Classification result

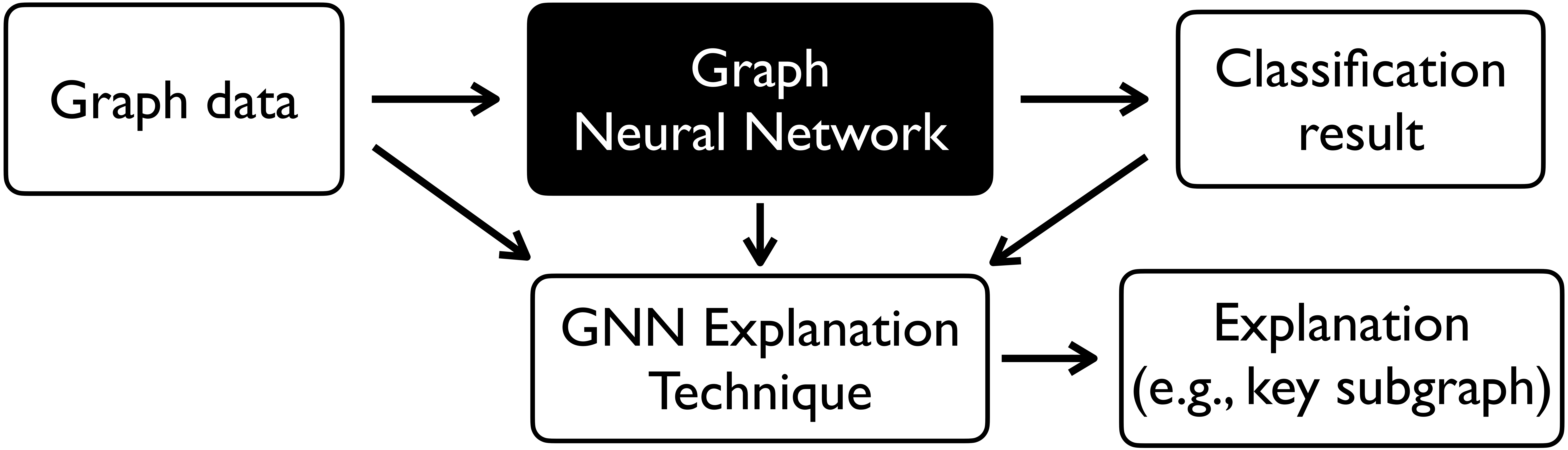
- Health care
- Trading activity
- …

Problem
Does not explain **why**

- Safe or not
- Fraud or not
- …

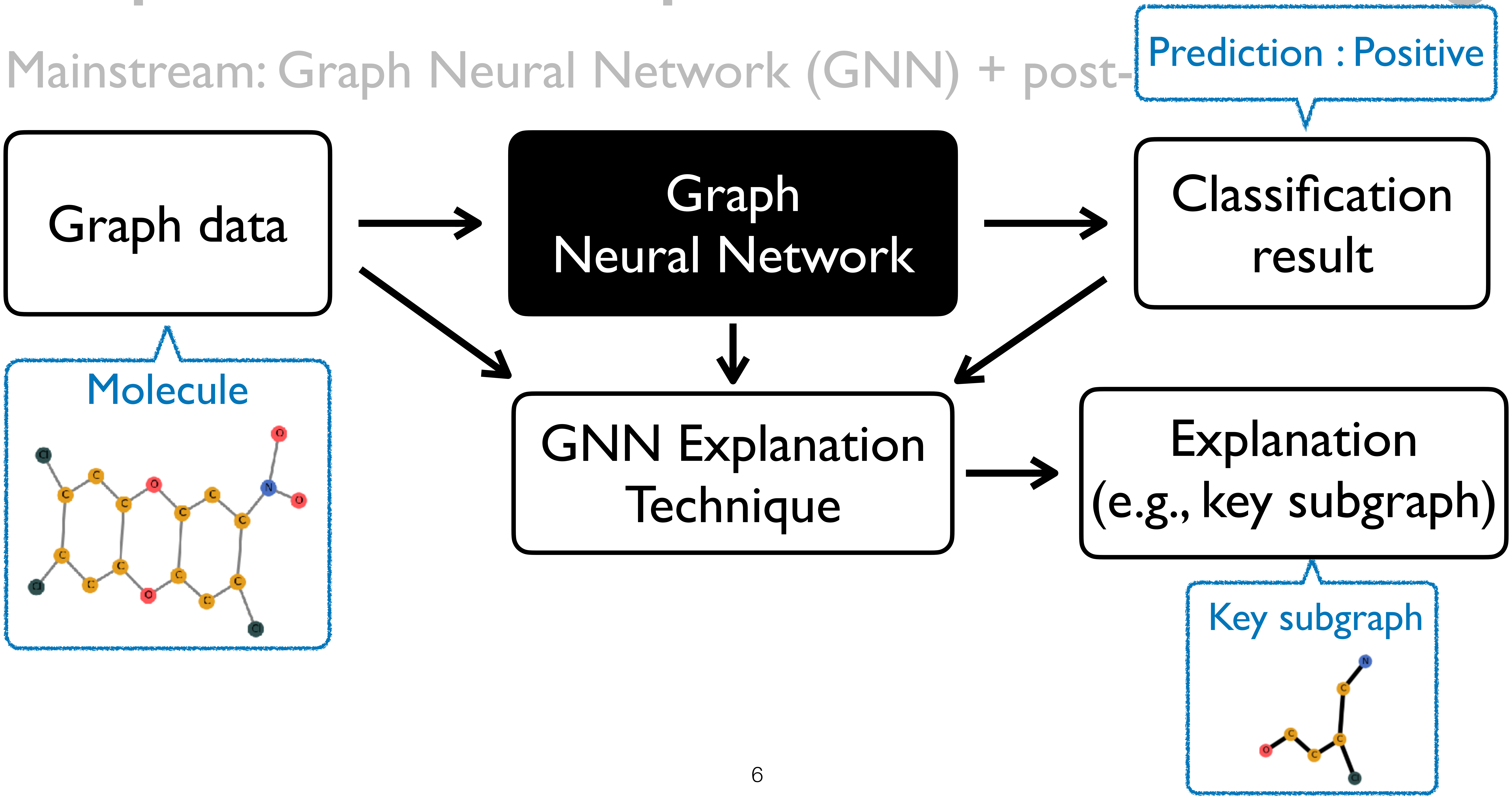
Explainable Graph Machine Learning

- Mainstream: Graph Neural Network (GNN) + post-hoc "explainers"



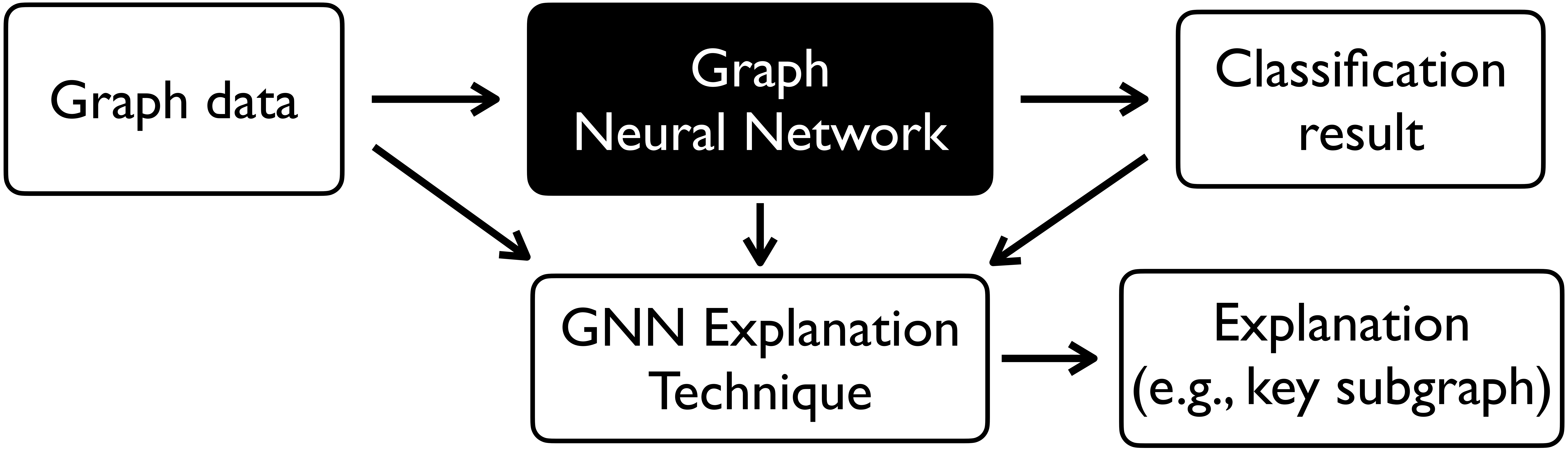
Explainable Graph Machine Learning

- Mainstream: Graph Neural Network (GNN) + post-



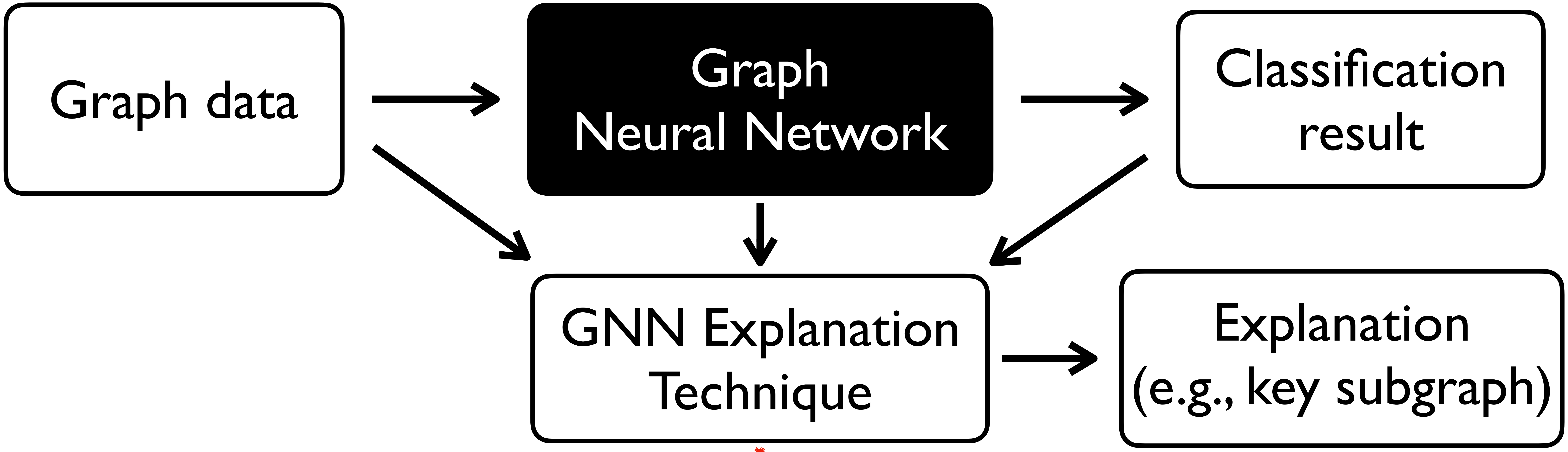
Explainable Graph Machine Learning

- Mainstream: Graph Neural Network (GNN) + post-hoc “explainers”



Explainable Graph Machine Learning

- Mainstream: Graph Neural Network (GNN) + post-hoc "explainers"



Two key limitations

- Additional (expensive) explanation cost is required
- The explanations are not guaranteed to be correct

Our Approach

- PL4XGL: PL-based inherently explainable graph machine learning method

Graph data → PL4XGL → Classification result & correct explanation

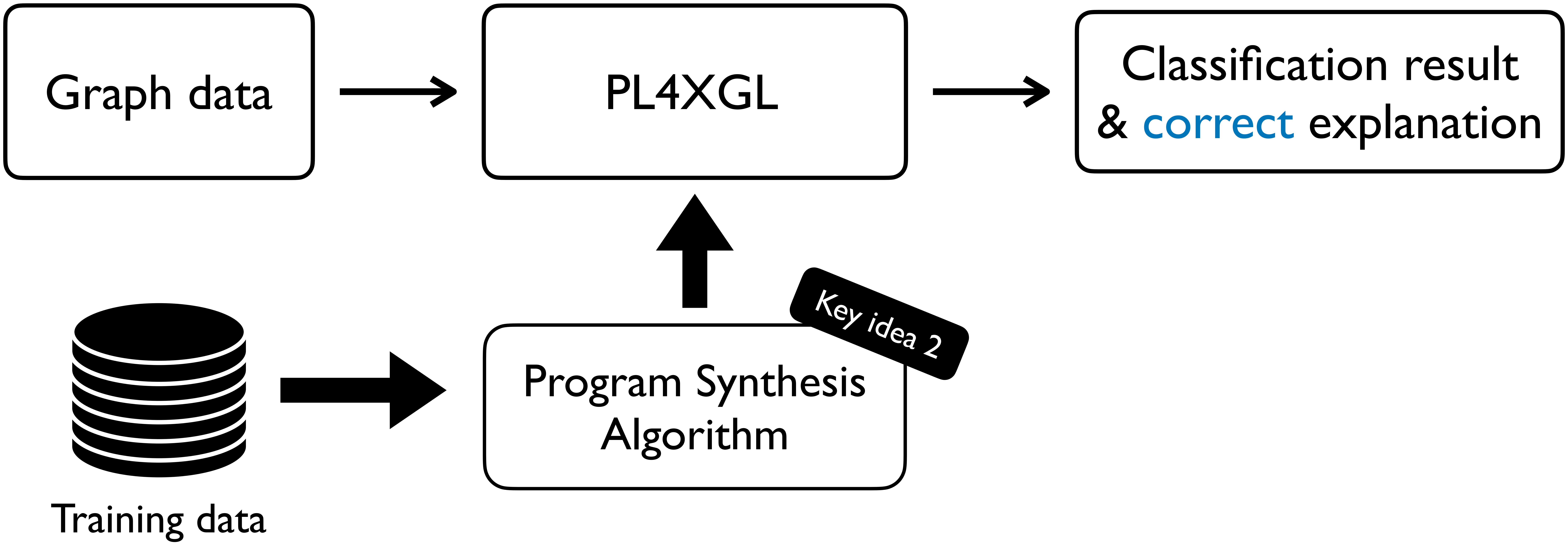
Key idea 1

Graph Description Language (GDL)

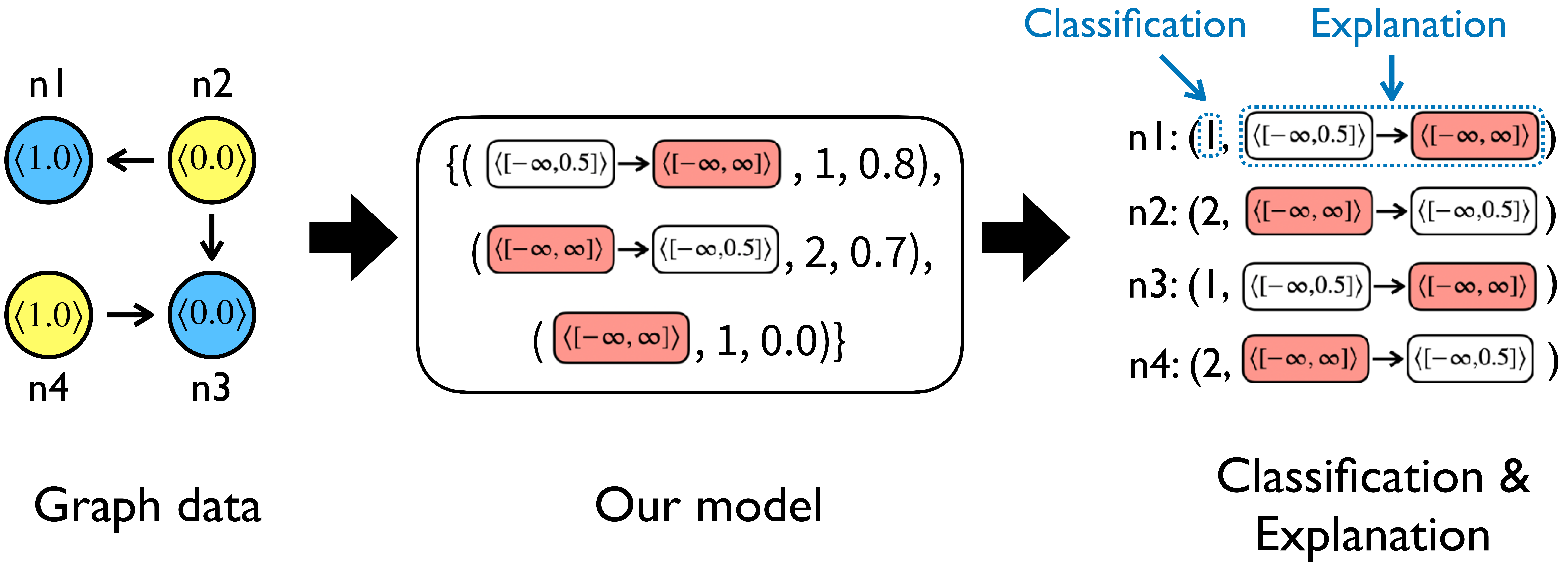
Programs	P	$::= \overline{\delta}$ **target** t	$\in$	$\mathbb{P} = \mathbb{D}^* \times \mathbb{T}$
Descriptions	δ	$::= \delta_V \mid \delta_E$	$\in$	$\mathbb{D} = \mathbb{D}_V \uplus \mathbb{D}_E$
Node Descriptions	δ_V	$::=$ **node** x $<\overline{\phi}>^?$	$\in$	$\mathbb{D}_V = \mathbb{X} \times \Phi^d$
Edge Descriptions	δ_E	$::=$ **edge** (x,x) $<\overline{\phi}>^?$	$\in$	$\mathbb{D}_E = \mathbb{X} \times \mathbb{X} \times \Phi^c$
Target Symbols	t	$::=$ **node** x $\mid$ **edge** (x,x) $\mid$ **graph**	$\in$	$\mathbb{T} = \mathbb{X} \uplus (\mathbb{X} \times \mathbb{X}) \uplus \{\epsilon\}$
Intervals	ϕ	$::= [n^?, n^?]$	$\in$	$\Phi = (\mathbb{R} \uplus \{-\infty\}) \times (\mathbb{R} \uplus \{\infty\})$
Real Numbers	n	$::= 0.2 \mid 0.7 \mid 6 \mid -8 \ \ldots$	$\in$	$\mathbb{R}$
Variables	x	$::= \mathtt{x} \mid \mathtt{y} \mid \mathtt{z} \mid \ldots$	$\in$	$\mathbb{X}$

Our Approach

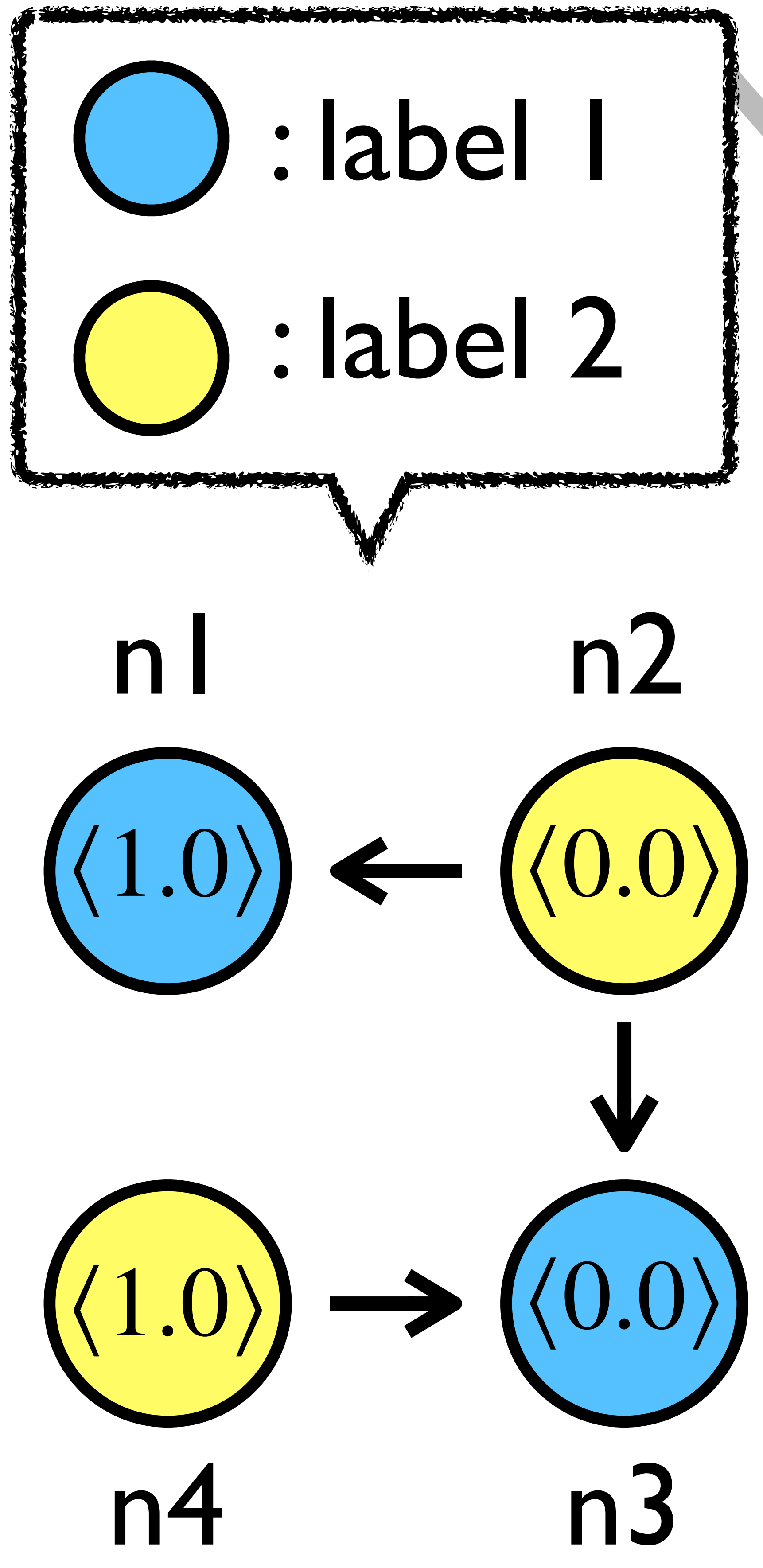
- PL4XGL: PL-based inherently explainable graph machine learning method



Node Classification Example



Node Classification Example



Graph data

$$\{(\boxed{\langle[-\infty,0.5]\rangle} \rightarrow \boxed{\langle[-\infty,\infty]\rangle}, 1, 0.8),$$
$$(\boxed{\langle[-\infty,\infty]\rangle} \rightarrow \boxed{\langle[-\infty,0.5]\rangle}, 2, 0.7),$$
$$(\boxed{\langle[-\infty,\infty]\rangle}, 1, 0.0)\}$$

Our model

Classification Explanation

n1: $(1, \langle[-\infty,0.5]\rangle \rightarrow \langle[-\infty,\infty]\rangle)$

n2: $(2, \langle[-\infty,\infty]\rangle \rightarrow \langle[-\infty,0.5]\rangle)$

n3: $(1, \langle[-\infty,0.5]\rangle \rightarrow \langle[-\infty,\infty]\rangle)$

n4: $(2, \langle[-\infty,\infty]\rangle \rightarrow \langle[-\infty,0.5]\rangle)$

Classification & Explanation

Node Classification Example

Graph data

$\{(\langle[-\infty,0.5]\rangle \rightarrow \langle[-\infty,\infty]\rangle, 1, 0.8),$
$(\langle[-\infty,\infty]\rangle \rightarrow \langle[-\infty,0.5]\rangle, 2, 0.7),$
$(\langle[-\infty,\infty]\rangle, 1, 0.0)\}$

Our model

Classification

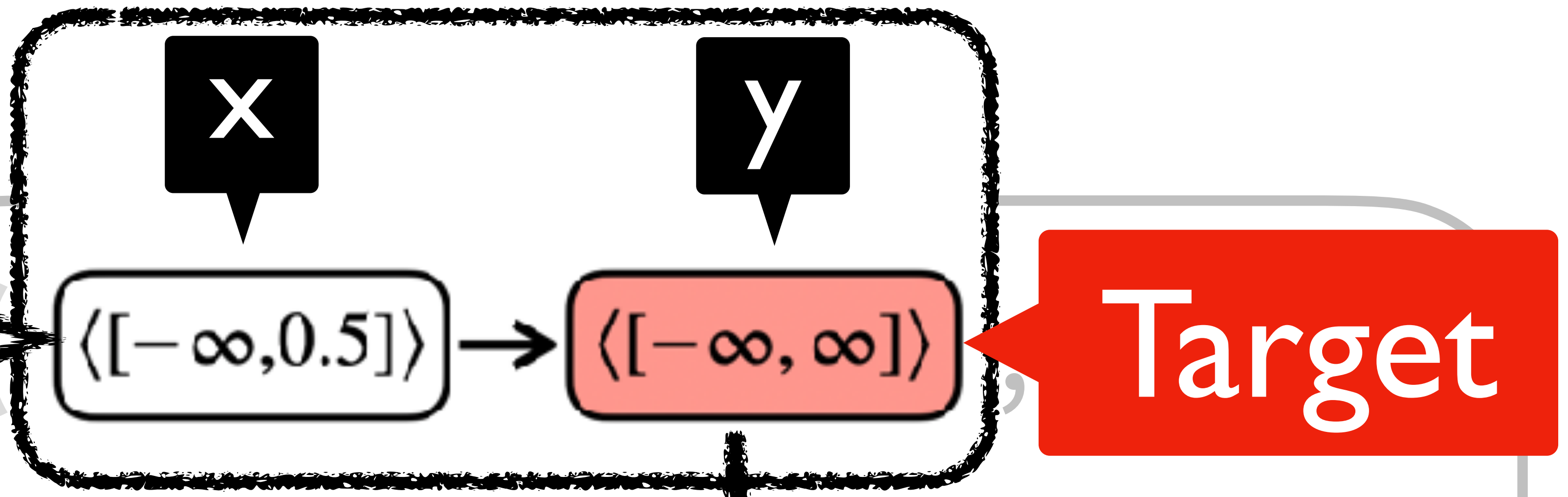
Explanation

n1: $(1, \langle[-\infty,0.5]\rangle \rightarrow \langle[-\infty,\infty]\rangle)$
n2: $(2, \langle[-\infty,\infty]\rangle \rightarrow \langle[-\infty,0.5]\rangle)$
n3: $(1, \langle[-\infty,0.5]\rangle \rightarrow \langle[-\infty,\infty]\rangle)$
n4: $(2, \langle[-\infty,\infty]\rangle \rightarrow \langle[-\infty,0.5]\rangle)$

Classification & Explanation

Node Classification Example

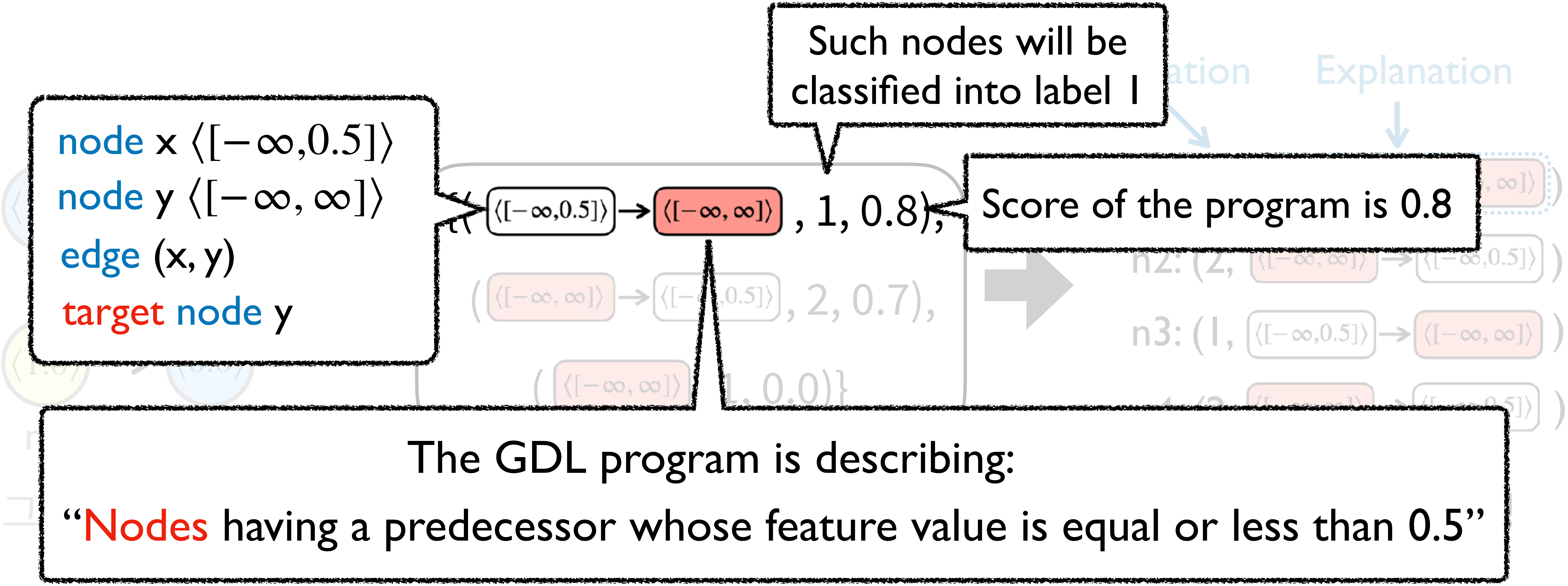
node x $\langle[-\infty, 0.5]\rangle$
node y $\langle[-\infty, \infty]\rangle$
edge (x, y)
target node y

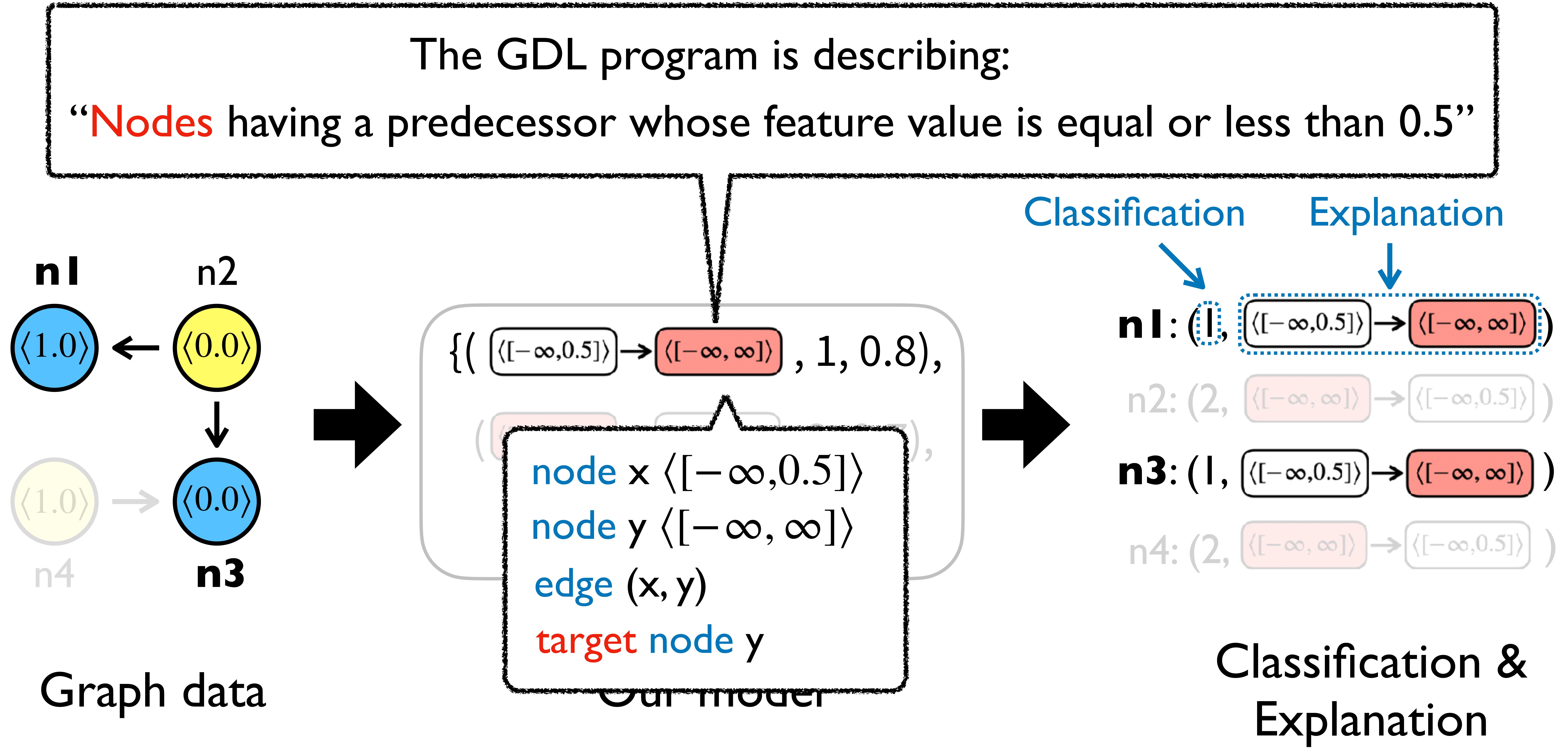


The GDL program is describing:

“Nodes having a predecessor whose feature value is equal or less than 0.5”

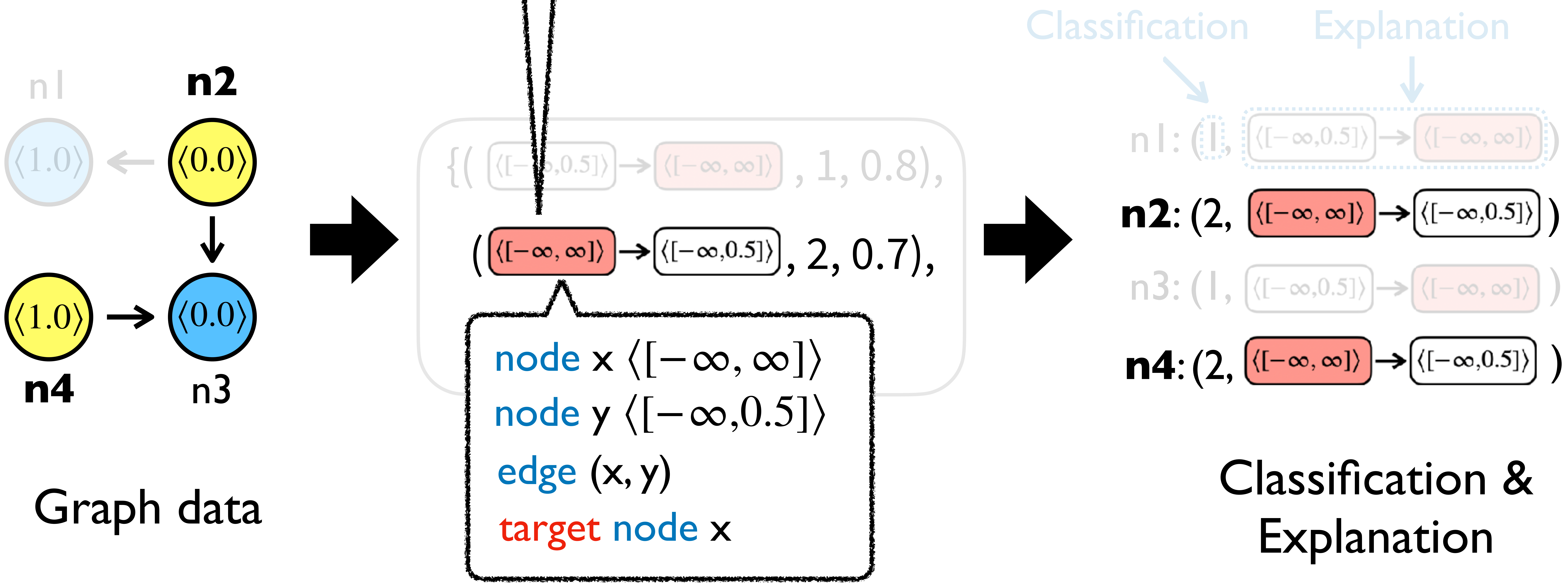
Node Classification Example



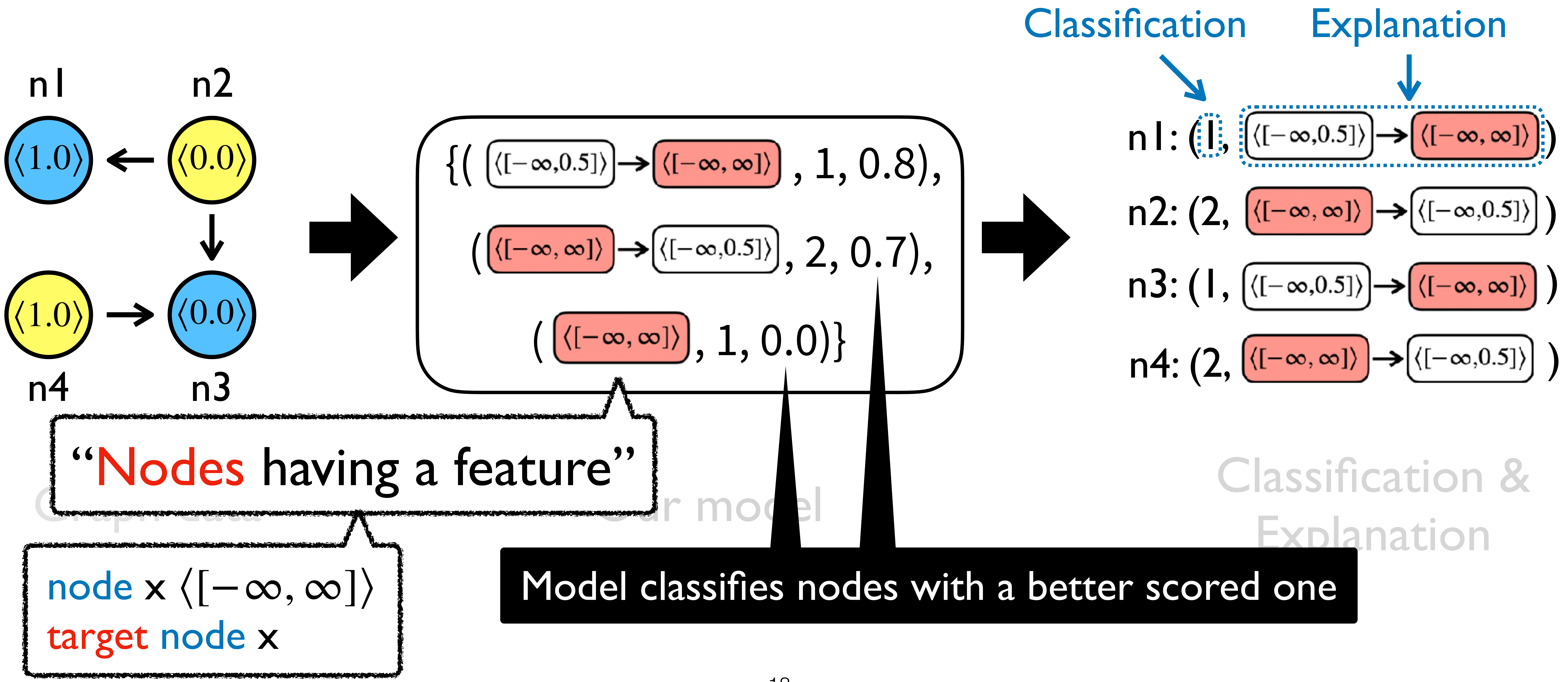

The GDL program is describing:
“Nodes having a predecessor whose feature value is equal or less than 0.5”
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
Graph data
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.8),
node x ⟨[−∞,0.5]⟩
node y ⟨[−∞, ∞]⟩
edge (x, y)
target node y
Classification
Explanation
n1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
Classification &
Explanation

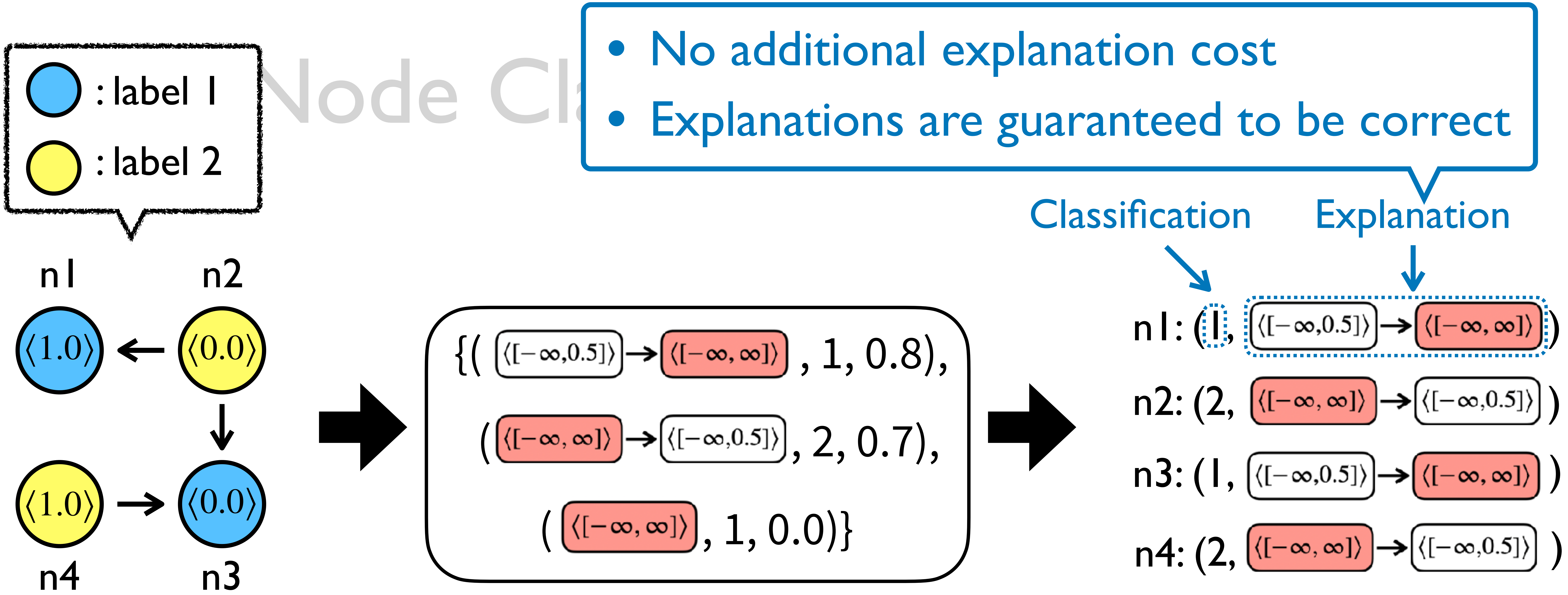
The GDL program is describing:

"Nodes having a successor whose feature value is equal or less than 0.5"



Node Classification Example



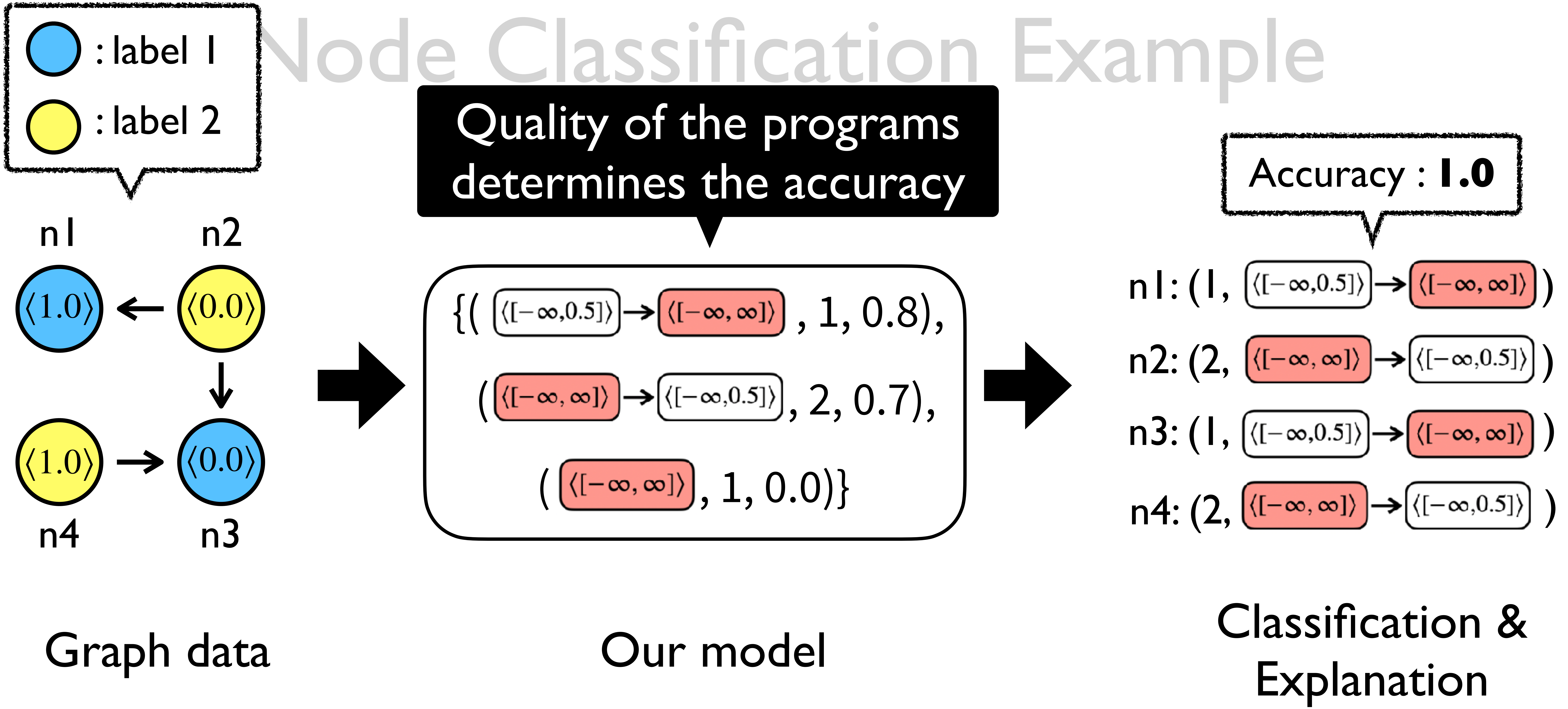


Graph data

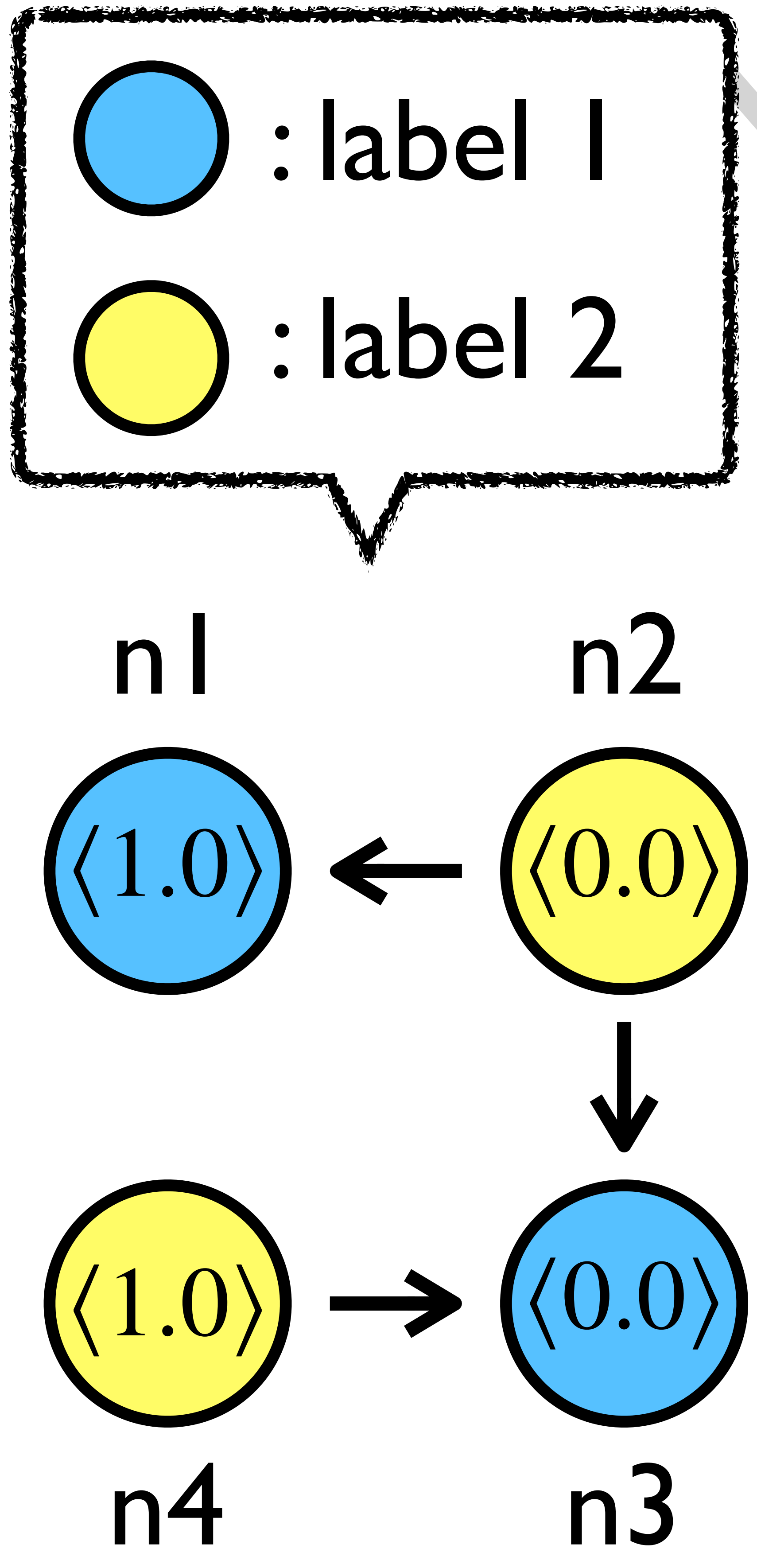
Our model

Classification & Explanation

Node Classification Example



Node Classification Example



Graph data

Quality of the programs determines the accuracy

$$\{(\langle[-\infty,\infty]\rangle \rightarrow \langle[-\infty,0.5]\rangle, 1, 0.8), (\langle[-\infty,0.5]\rangle \rightarrow \langle[-\infty,\infty]\rangle, 2, 0.7), (\langle[-\infty,\infty]\rangle, 1, 0.0)\}$$

Our model

Accuracy : **0.0**

n1: (**2**, $\langle[-\infty,0.5]\rangle \rightarrow \langle[-\infty,\infty]\rangle$)

n2: (**1**, $\langle[-\infty,\infty]\rangle \rightarrow \langle[-\infty,0.5]\rangle$)

n3: (**2**, $\langle[-\infty,0.5]\rangle \rightarrow \langle[-\infty,\infty]\rangle$)

n4: (**1**, $\langle[-\infty,\infty]\rangle \rightarrow \langle[-\infty,0.5]\rangle$)

Classification & Explanation


Training data
Learning algorithm
Learning objective:
Learn high-quality GDL programs
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
{(⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩, 1, 0.8),
(⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
Graph data
Our model
Classification &
Explanation


Training data
Learning algorithm
Top-down synthesis algorithm
Bottom-up synthesis algorithm
Learning objective:
Learn high-quality GDL programs
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
{(⟨[−∞,0.5]⟩→⟨[−∞,∞]⟩, 1, 0.8),
(⟨[−∞,∞]⟩→⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞,∞]⟩, 1, 0.0)}
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞,∞]⟩)
n2: (2, ⟨[−∞,∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞,∞]⟩)
n4: (2, ⟨[−∞,∞]⟩→⟨[−∞,0.5]⟩)
Graph data
Our model
Classification & Explanation

Evaluation

- Compared PL4XGL with
  - Representative GNNs : GCN, GAT, GIN, etc
  - State-of-the-art GNN explainer : SubgraphX*
- Research questions:
  - RQ1) Classification accuracy
  - RQ2) Explainability
- Settings:
  - GNNs and SubgraphX trained and evaluated using a GPU (RTX A6000)
  - PL4XGL trained and evaluated using 64-core CPU

---

*Yuan et al. On explainability of graph neural networks via subgraph explorations. ICML 2021

RQ1) Classification Accuracy

- Each dataset is split into 8:1:1 for training, validation, and evaluation
- PL4XGL achieved the best accuracy for 5 datasets
- PL4XGL did not scale for the largest dataset HIV (time budget = 48h)

	GCN	GAT	ChebyNet	JKNet	GraphSage	GIN	DGCN	PL4XGL
MUTAG	80.0±0.0	89.0±2.2	86.0±4.1	68.0±7.5	78.0±4.4	91.0±5.4	N/A	**100.0±0.0**
BBBP	83.6±1.4	82.3±1.6	84.6±1.0	85.6±1.9	86.6±0.9	86.2±1.4	N/A	**86.8±0.0**
BACE	78.4±2.8	52.4±3.3	78.9±1.4	79.9±1.9	79.8±0.8	**80.9±0.4**	N/A	**80.9±0.0**
HIV	96.4±0.0	96.4±0.0	96.8±0.2	96.8±0.1	**96.9±0.2**	96.8±0.1	N/A	N/A
BA-Shapes	95.1±0.6	76.8±2.3	**97.1±0.0**	94.3±0.0	**97.1±0.0**	92.0±1.1	95.1±0.7	95.7±0.0
Tree-Cycles	97.7±0.0	90.9±0.0	**100.0±0.0**	98.9±0.0	**100.0±0.0**	93.2±0.0	99.2±0.5	**100.0±0.0**
Wisconsin	64.0±0.0	49.6±3.1	86.4±3.9	64.8±1.5	92.8±2.9	56.0±0.0	**96.0±0.0**	88.0±0.0
Texas	67.7±5.3	50.0±0.0	87.7±2.1	68.8±4.3	86.6±2.6	50.0±0.0	**86.6±2.6**	83.3±0.0
Cornell	58.9±2.6	61.1±0.0	81.0±6.5	61.1±0.0	87.7±2.1	61.1±0.0	86.6±2.6	**88.8±0.0**
Cora	85.6±0.3	86.4±1.8	86.5±5.2	84.9±3.5	86.3±3.2	**86.7±0.0**	83.2±5.9	80.0± 0.0
Citeseer	75.2±0.0	74.3±0.7	**79.1±0.9**	73.7±4.2	75.9±2.3	75.2±0.0	71.3±6.0	63.8± 0.0
Pubmed	82.8±1.1	84.7±1.2	**88.7±1.0**	83.2±0.4	88.0±0.4	86.1±0.6	85.1±0.6	81.4±0.0

RQ1) Classification Accuracy

- Each dataset is split into 8:1:1 for training, validation, and evaluation
- PL4XGL
- PL4XGL did not scale for the largest dataset HIV (time budget = 48h)

Molecule datasets (graph classification)

	GCN	GAT	ChebyNet	JKNet	GraphSage	GIN	DGCN	PL4XGL
MUTAG	80.0±0.0	89.0±2.2	86.0±4.1	68.0±7.5	78.0±4.4	91.0±5.4	N/A	**100.0±0.0**
BBBP	83.6±1.4	82.3±1.6	84.6±1.0	85.6±1.9	86.6±0.9	86.2±1.4	N/A	**86.8±0.0**
BACE	78.4±2.8	52.4±3.3	78.9±1.4	79.9±1.9	79.8±0.8	**80.9±0.4**	N/A	**80.9±0.0**
HIV	96.4±0.0	96.4±0.0	96.8±0.2	96.8±0.1	**96.9±0.2**	96.8±0.1	N/A	N/A
BA-Shapes	95.1±0.6	76.8±2.3						
Tree-Cycles	97.7±0.0	90.9±0.0	1					
Wisconsin	64.0±0.0	49.6±3.1						
Texas	67.7±5.3	50.0±0.0						
Cornell	58.9±2.6	61.1±0.0	81.0±6.5	61.1±0.0	87.7±2.1	61.1±0.0	86.6±2.6	**88.8±0.0**
Cora	85.6±0.3	86.4±1.8	86.5±5.2	84.9±3.5	86.3±3.2	**86.7±0.0**	83.2±5.9	80.0± 0.0
Citeseer	75.2±0.0	74.3±0.7	**79.1±0.9**	73.7±4.2	75.9±2.3	75.2±0.0	71.3±6.0	63.8± 0.0
Pubmed	82.8±1.1	84.7±1.2	**88.7±1.0**	83.2±0.4	88.0±0.4	86.1±0.6	85.1±0.6	81.4±0.0

PL4XGL shows the best accuracy

RQ1) Classification Accuracy

- Each dataset is split into 8:1:1 for training, validation, and evaluation
- PL4XGL achieved the best accuracy for 5 datasets
- PL4XGL did not scale for the largest dataset HIV (time budget = 48h)

	GCN	GAT	ChebyNet	JKNet	GraphSage	GIN	DGCN	PL4XGL
MUTAG	80.0±0.0	89.0±2.2	86.0±4.1	68.0±7.5	78.0±4.4	91.0±5.4	N/A	**100.0±0.0**
BBBP	83.6±1.4	82.3±1.6	84.6±1.0	85.6±1.9	86.6±0.9	86.2±1.4	N/A	**86.8±0.0**
BACE	78.4±2.8	52.4±3.3	78.9±1.4	79.9±1.9	79.8±0.8	**80.9±0.4**	N/A	**80.9±0.0**
HIV	96.4±0.0	96.4±0.0	96.8±0.2	96.8±0.1	**96.9±0.2**	96.8±0.1	N/A	N/A
BA-Shapes	95.1±0.6	76.8±2.3	**97.1±0.0**	94.3±0.0	**97.1±0.0**	92.0±1.1	95.1±0.7	95.7±0.0
Pubmed	82.8±1.1	84.7±1.2	**88.7±1.0**	83.2±0.4	88.0±0.4	86.1±0.6	85.1±0.6	81.4±0.0

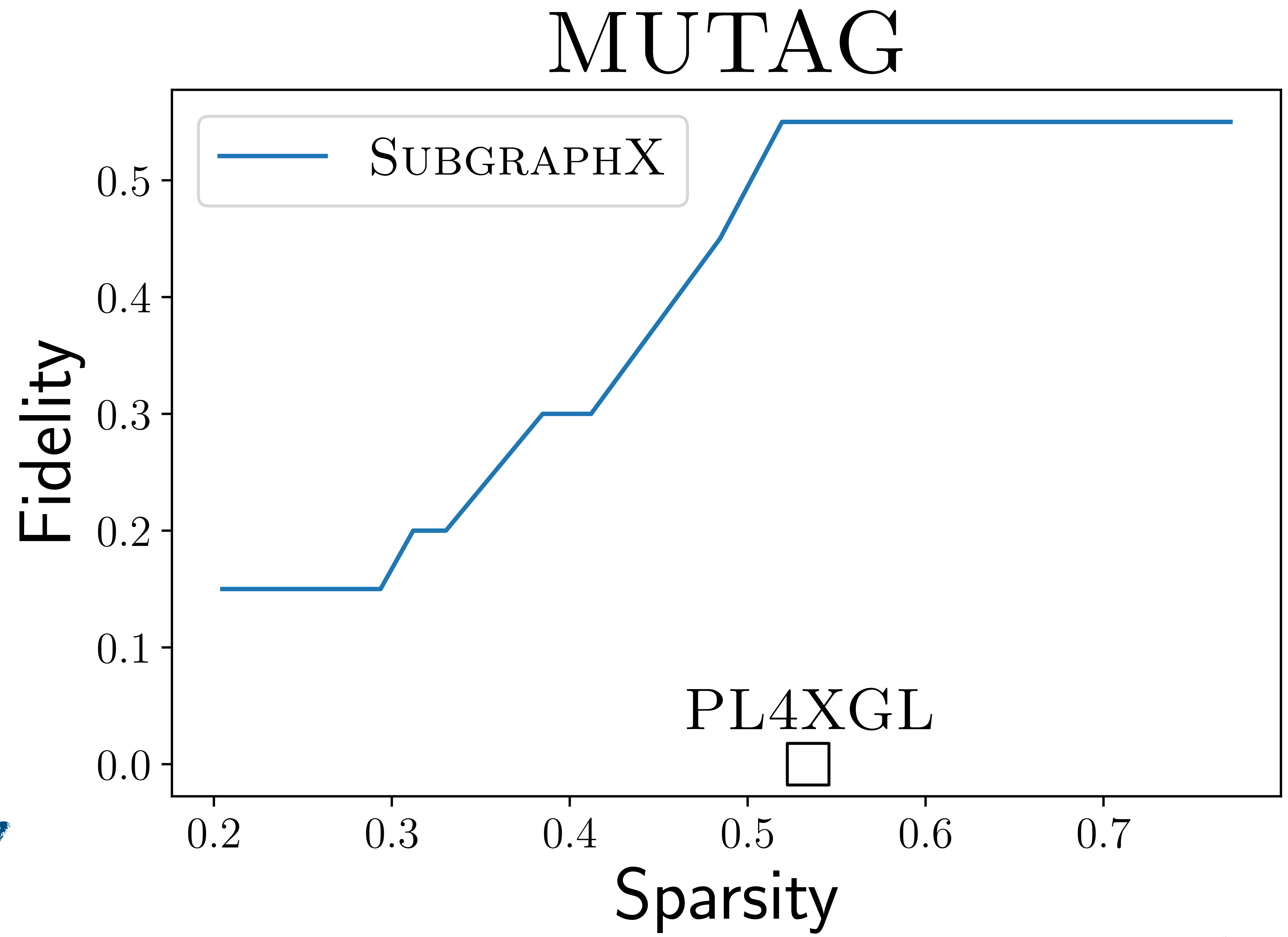
PL4XGL failed its training in HIV dataset because of its training cost

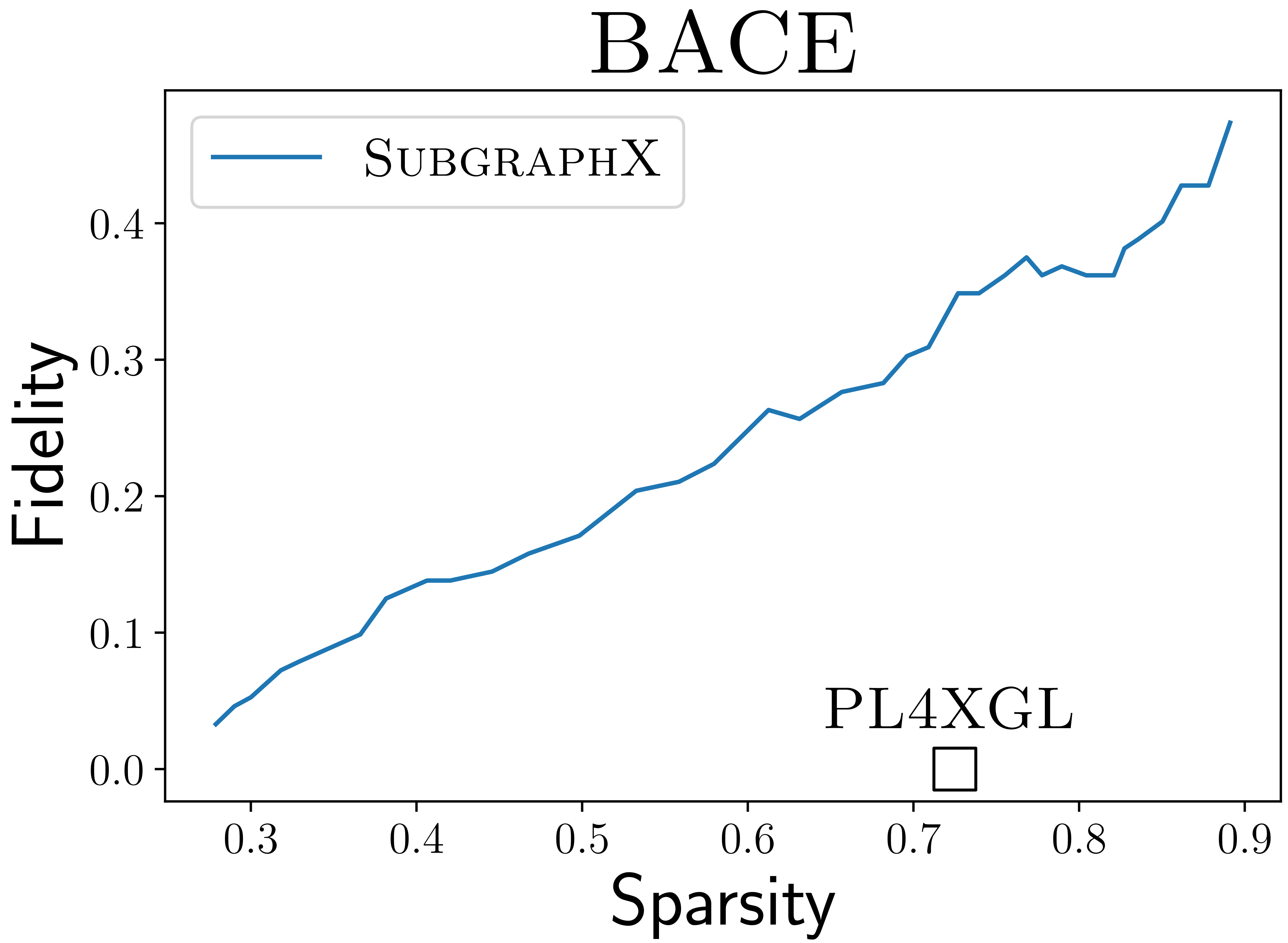
- HIV includes 41,127 (1,049,163 nodes)
- Timeout = 2 day (48 hours)

RQ2) Explainability

- Our approach provides correct & simple explanations

Explanations are correct

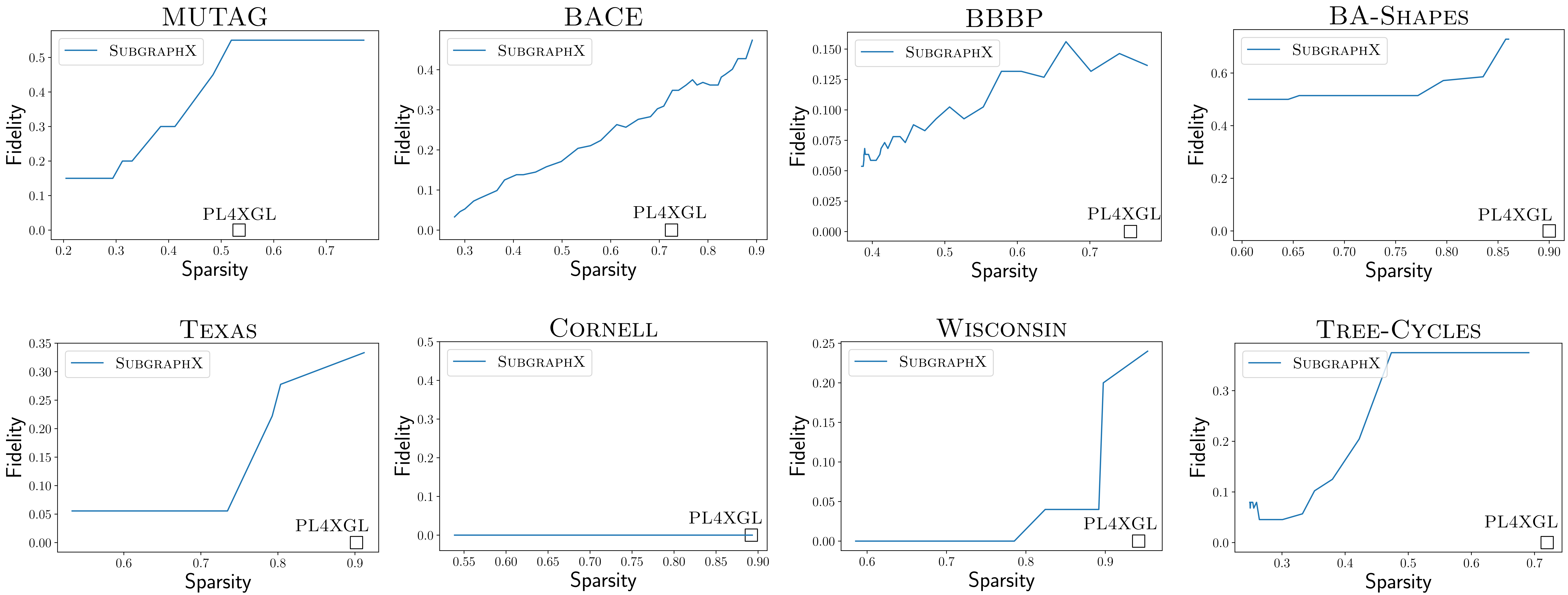




The explanations are simple

RQ2) Explainability

- Our approach provides correct & simple explanations



Summary

- Problem : Accurate and explainable graph learning
- Solution : A purely PL-based approach to XAI
  - Domain specific language design for defining AI models
  - Program synthesis for learning models from training data
- Result:
  - Accuracy can compete with GNNs
  - Better explainability than GNNs with post-hoc explainer

Summary

- Problem : Accurate and explainable graph learning
- Solution : A purely PL-based approach to XAI
  - Domain specific language design for defining AI models
  - Program synthesis for learning models from training data
- Result:
  - Accuracy can compete with GNNs
  - Better explainability than GNNs with post-hoc explainer

Conclusion: PL techniques are even useful for AI!