PL4XGL: 프로그래밍 언어 기법을 활용한 설명 가능한 그래프 기계학습 방법

PLDI 2024

전민석

고려대학교

05.03.2024@Prosys Lab

- 화학반응 예측
- 사기 거래 탐지
- 헬스 케어
- 프로그램 분석
- …

- 양성 / 음성
- 탐지 됨 / 탐지 안됨
- 부작용 있음 / 없음
- 버그가 있음 / 없음
- …

그래프 데이터 → 기계학습 모델 → 분류 & 예측

arXiv

https://arxiv.org › cs

Semi-Supervised Classification with Graph Convolutional ...

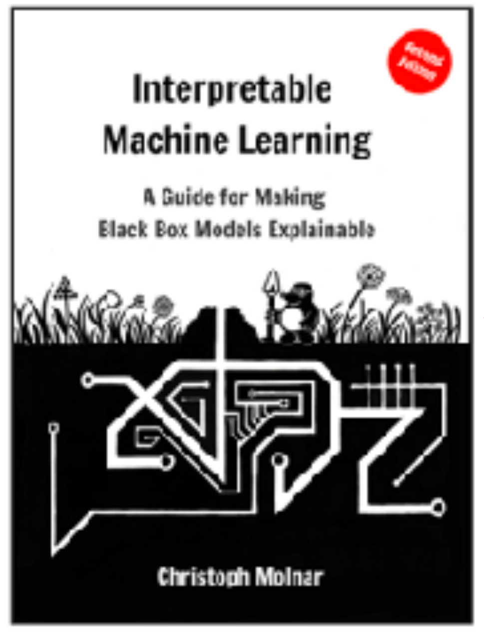
TN Kipf 저술 2016 · 33445회 인용 — We present a scalable approach for **semi-supervised** learning on **graph**-structured data that is based on an efficient variant of **convolutional** ...

그래프 데이터 → GNN (Graph Neural Network) → 분류 & 예측

그래프 데이터 → GNN (Graph Neural Network) → 분류 & 예측

문제점
예측의 이유를 설명해 주지 않음

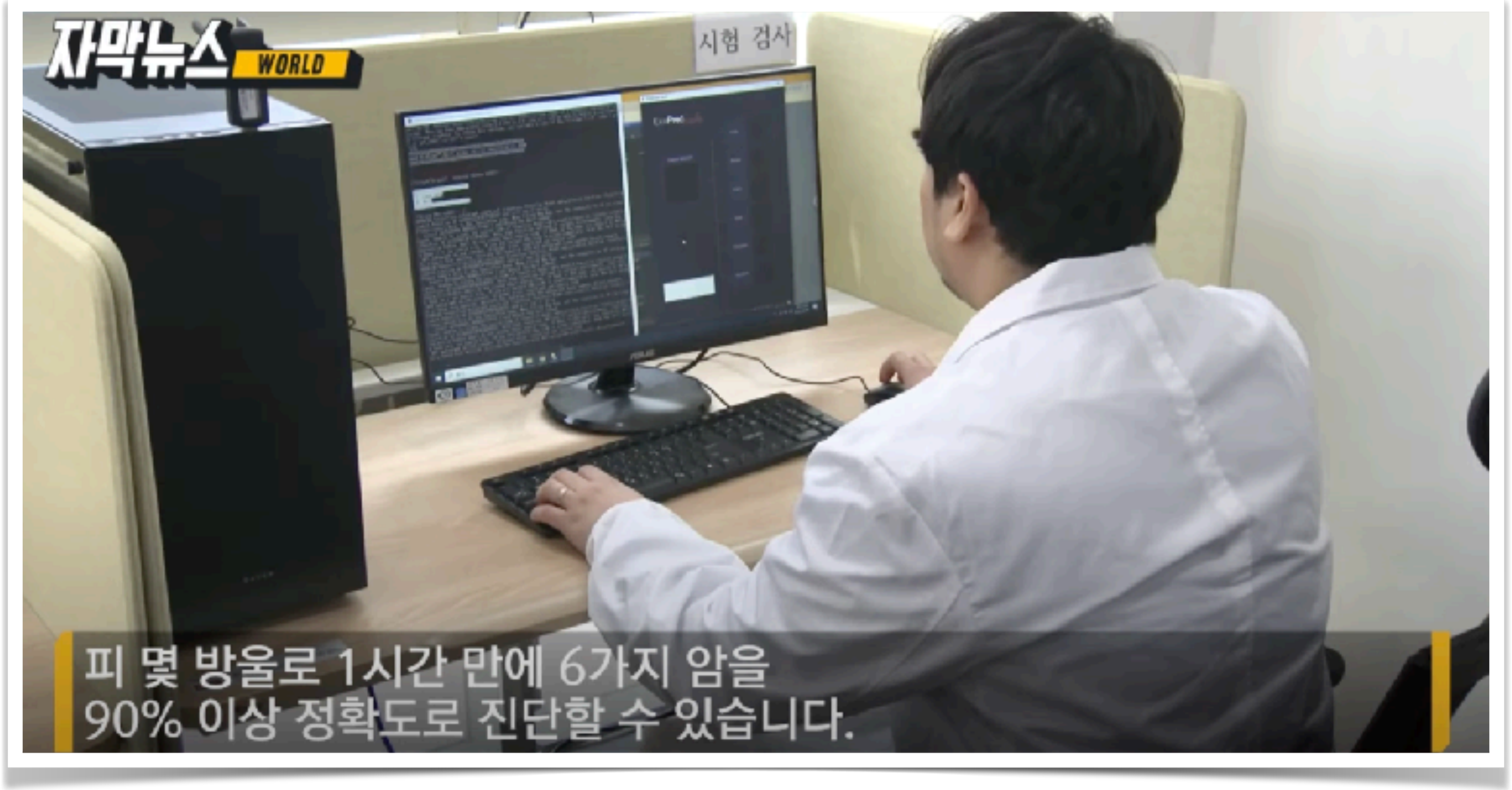
예측의 **이유**에 대한 수요가 매우 큼

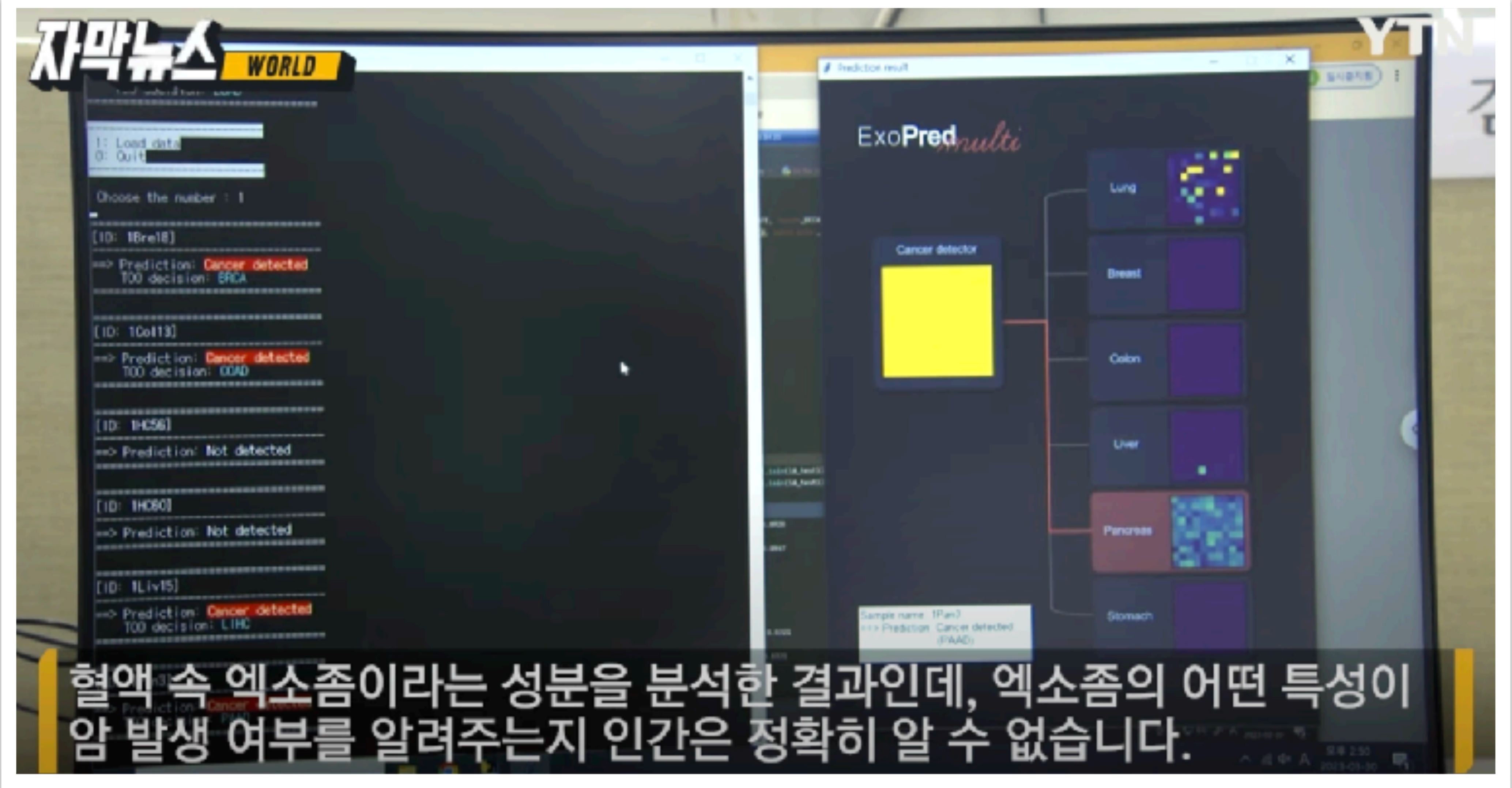


A correct prediction only partially solves your problem. The model must also explain **why**.

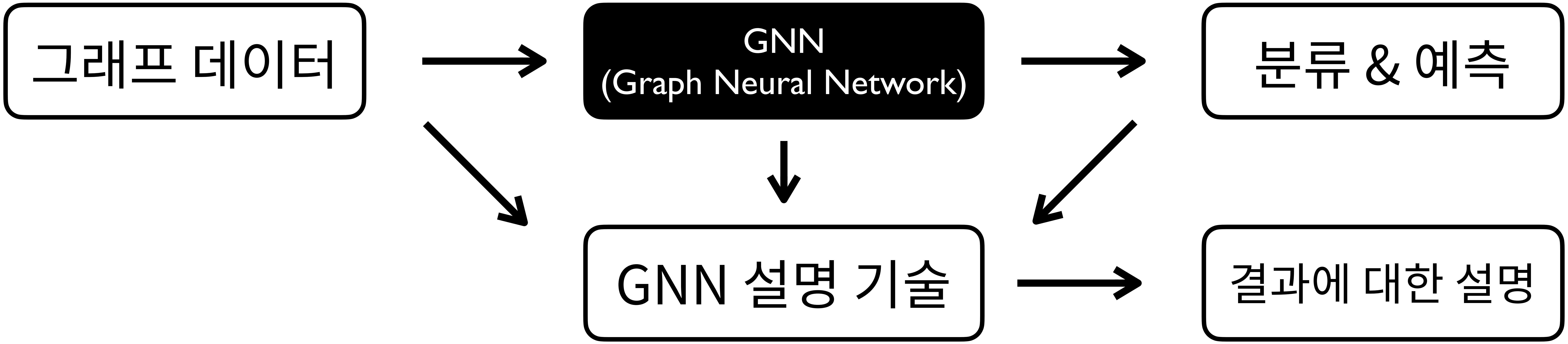
- Molnar [2022]

설명 가능한 기계학습 방법의 필요성





https://www.youtube.com/watch?v=DFT3P-9G06U

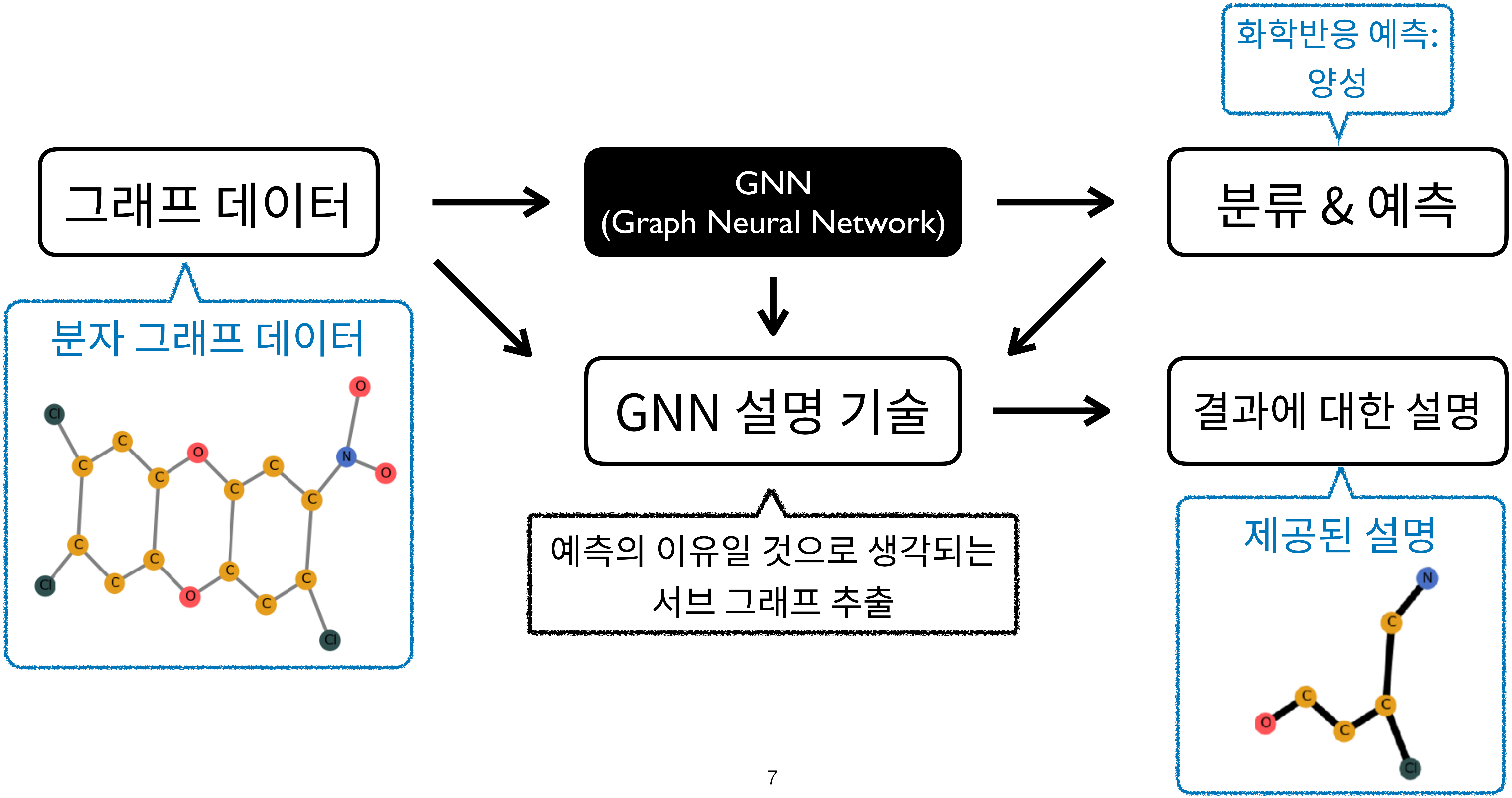


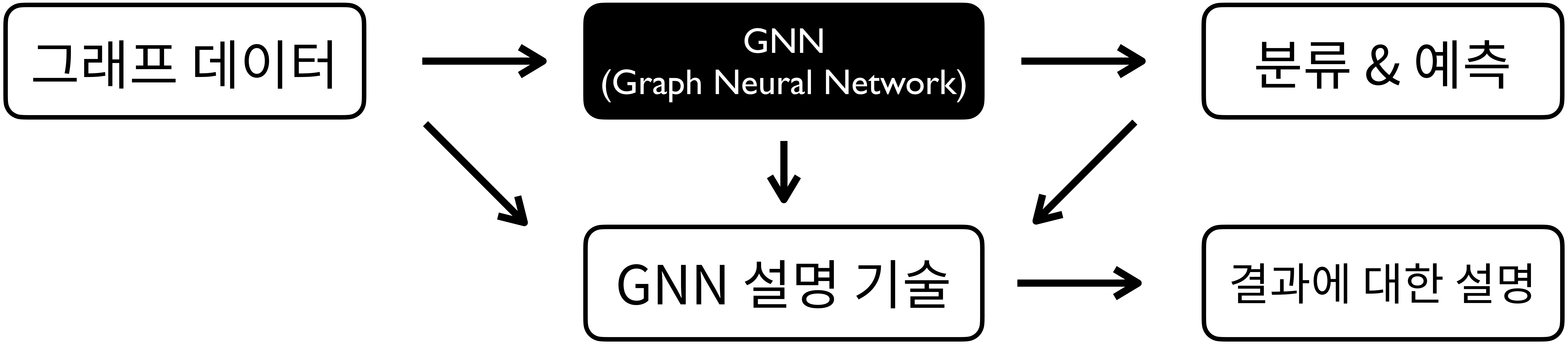
수 많은 GNN 설명 기술들이 개발되는 중

arXiv
https://arxiv.org › cs

GNNExplainer: Generating Explanations for Graph Neural ...

R Ying 저술 · 2019 · 1191회 인용 — Here we propose **GNNExplainer**, the first general, model-agnostic approach for providing interpretable explanations for predictions of any GNN- ...

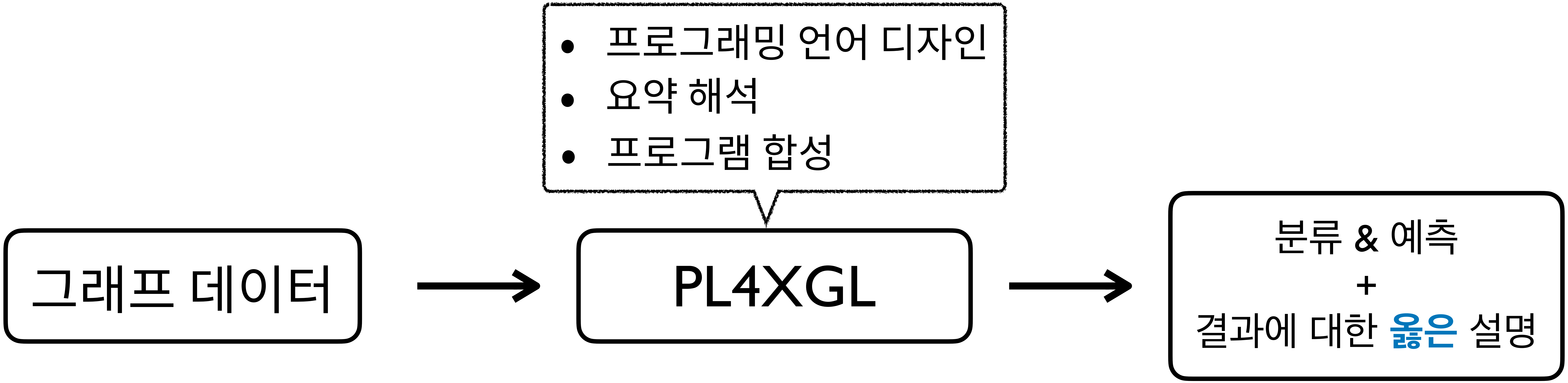

화학반응 예측:
양성
그래프 데이터
GNN
(Graph Neural Network)
분류 & 예측
분자 그래프 데이터
GNN 설명 기술
결과에 대한 설명
예측의 이유일 것으로 생각되는
서브 그래프 추출
제공된 설명

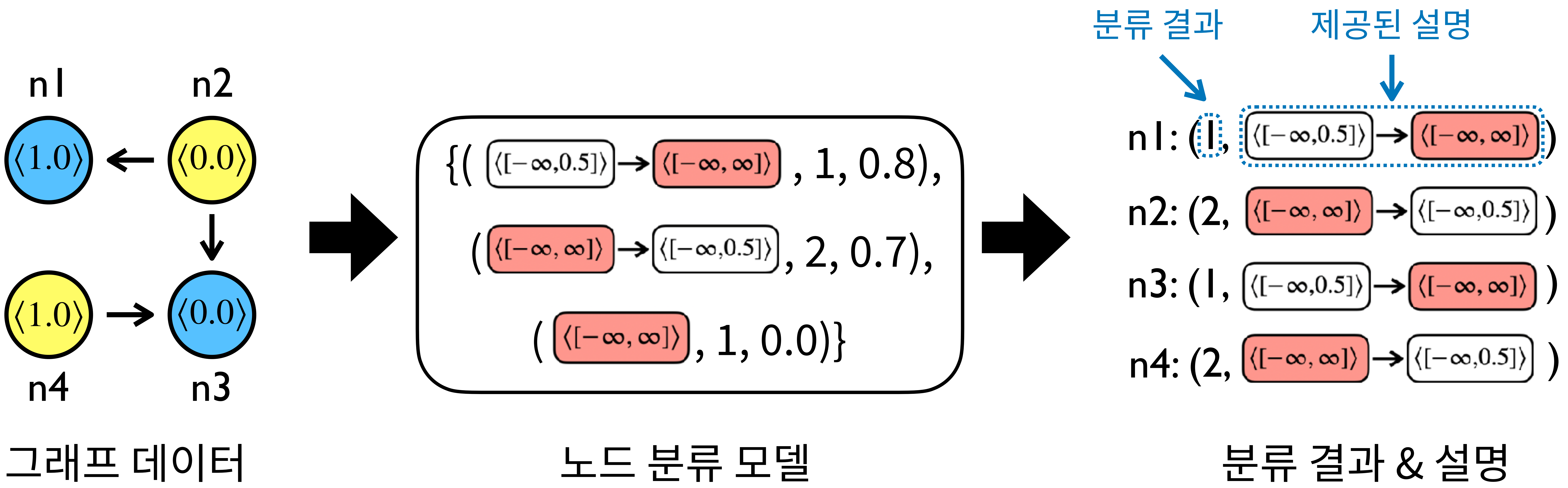


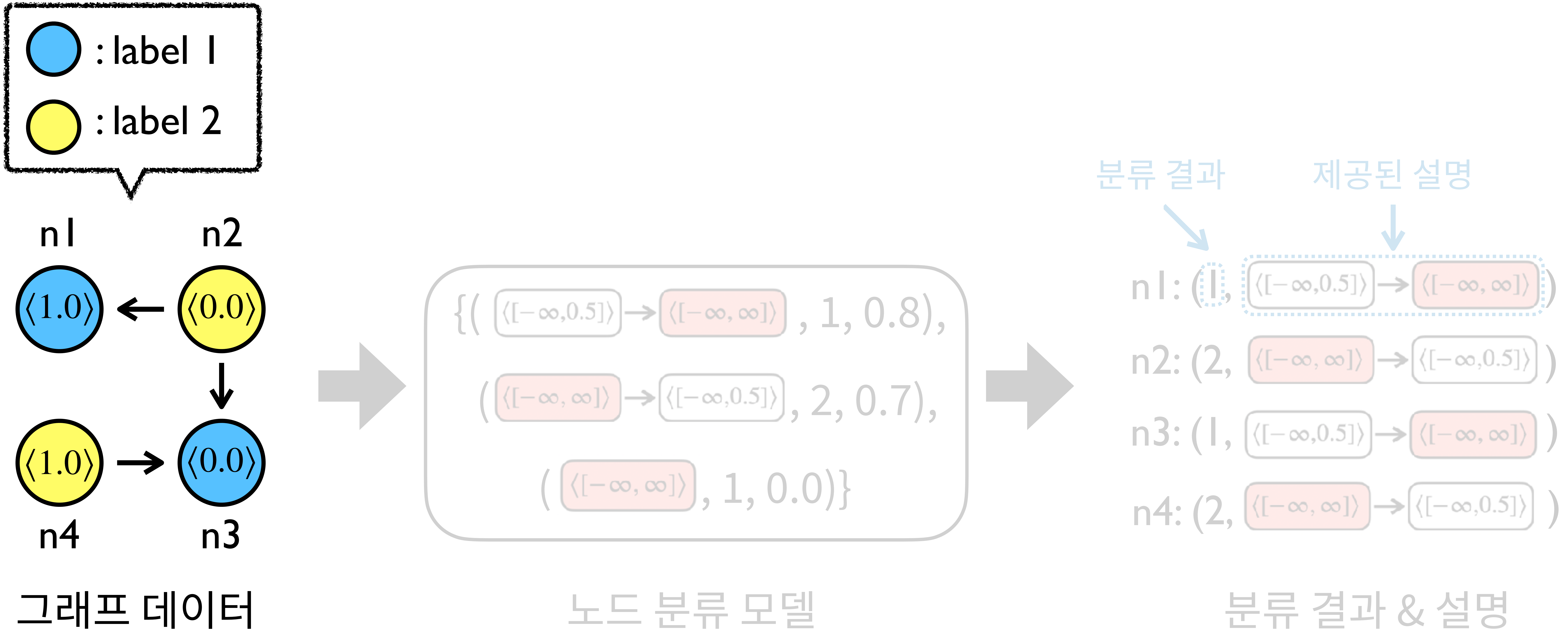
GNN 설명 방법들의 두 가지 핵심 한계

(1) 추가적인 (**비싼**) 설명 비용이 필수적

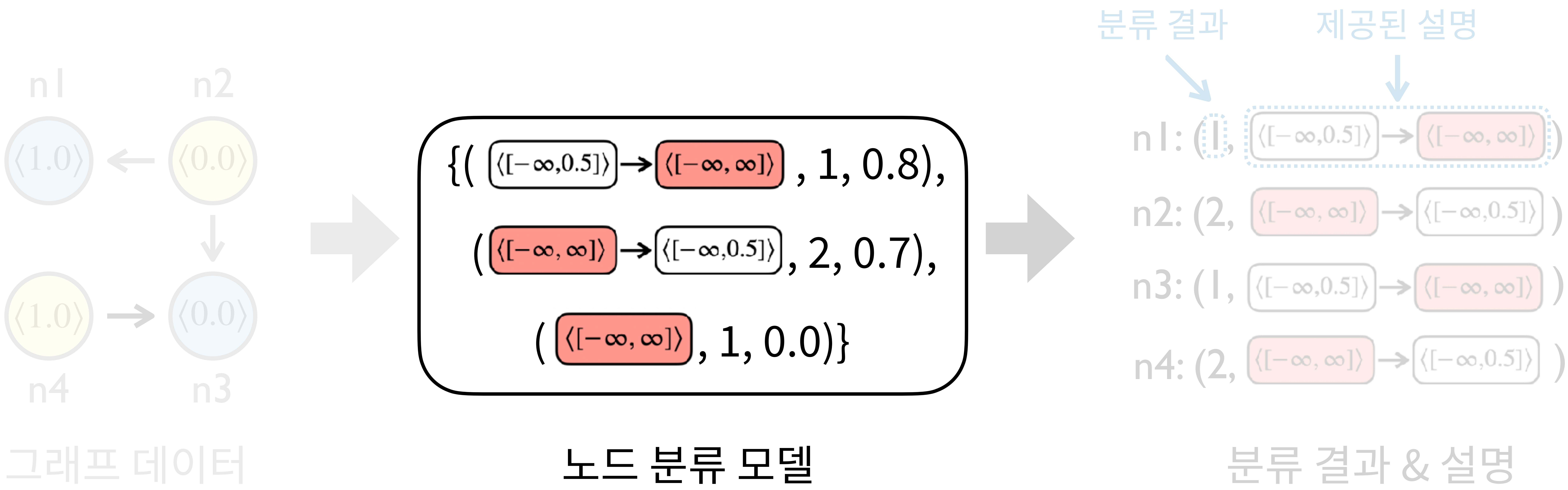
(2) 제공된 설명이 옳은 설명임을 보장해 주지 않음


• 프로그래밍 언어 디자인
• 요약 해석
• 프로그램 합성
그래프 데이터
PL4XGL
분류 & 예측
+
결과에 대한 옳은 설명

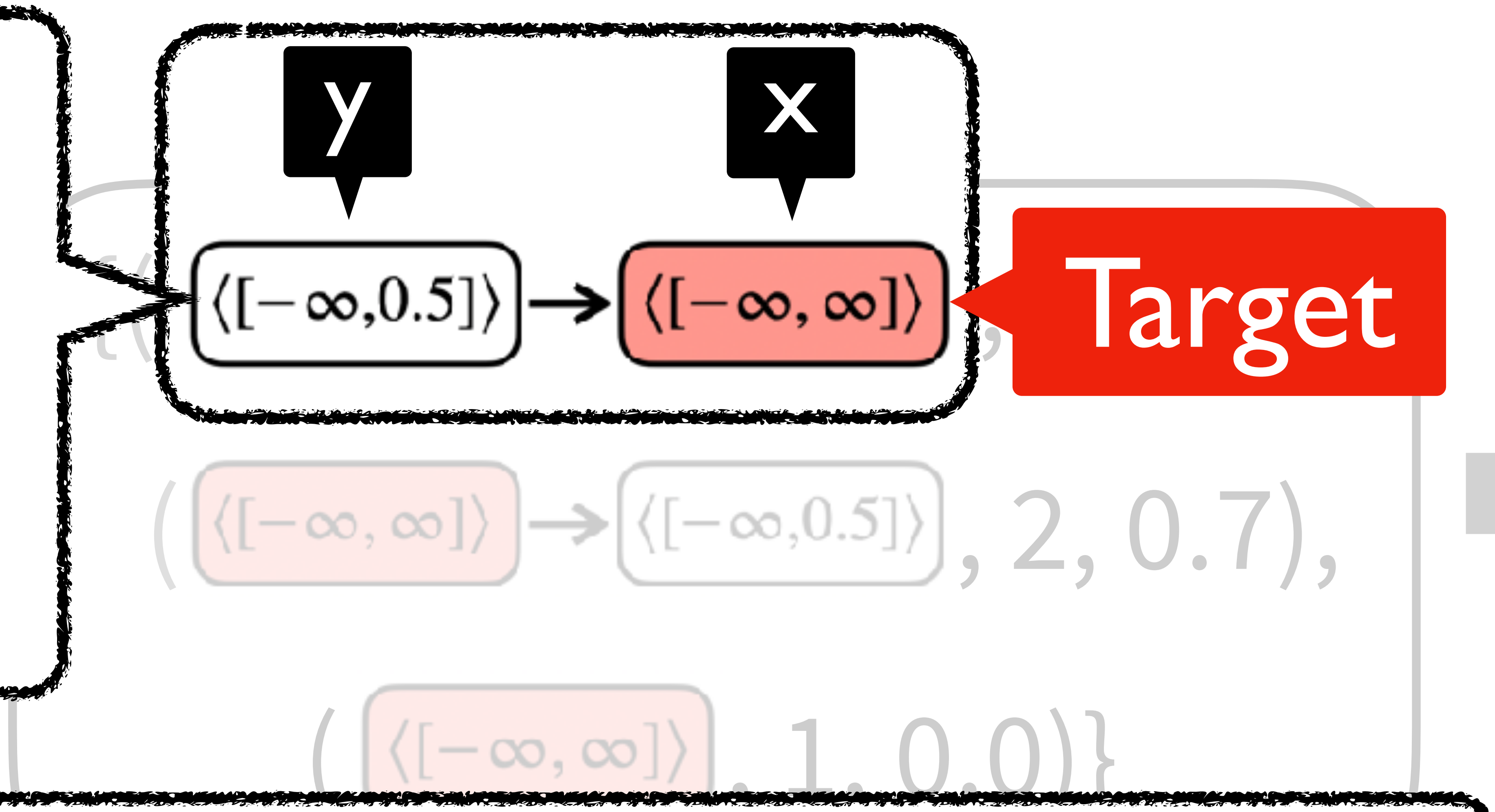

분류 결과
제공된 설명
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩ , 2, 0.7),
(⟨[−∞, ∞]⟩ , 1, 0.0)}
n1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
그래프 데이터
노드 분류 모델
분류 결과 & 설명

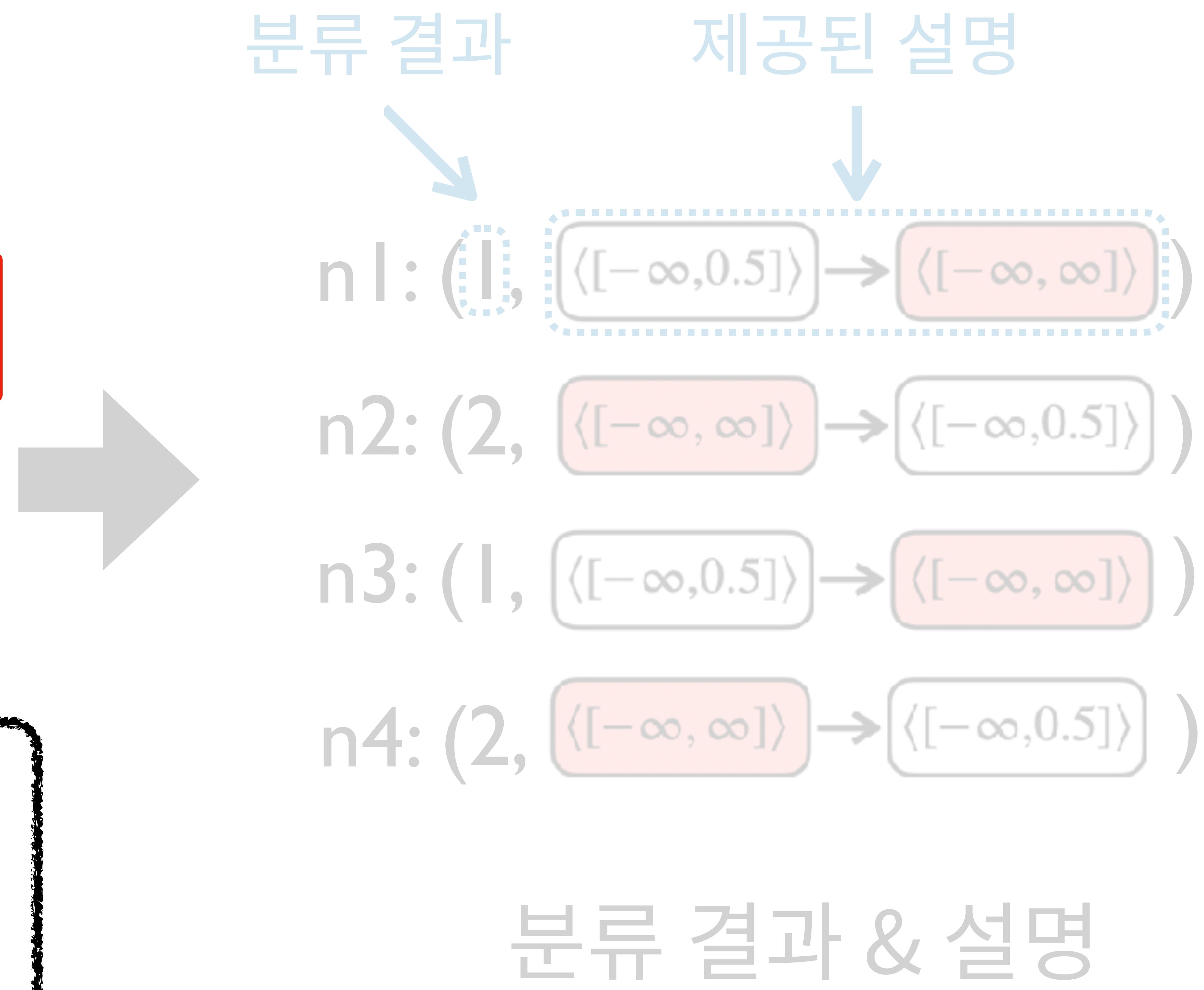


: label 1
: label 2
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
그래프 데이터
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
노드 분류 모델
분류 결과
제공된 설명
n1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
분류 결과 & 설명


n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
그래프 데이터
{(⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩, 1, 0.8),
(⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
노드 분류 모델
분류 결과
제공된 설명
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
분류 결과 & 설명

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

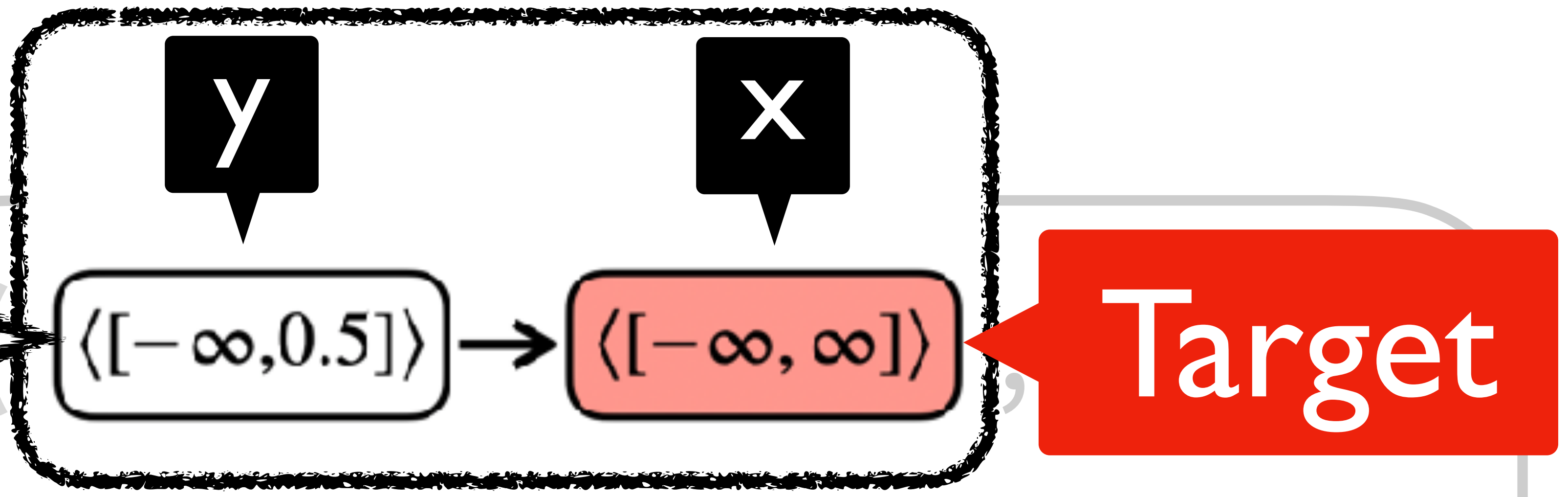




그래프 패턴 표현 언어 (Graph Description Language)

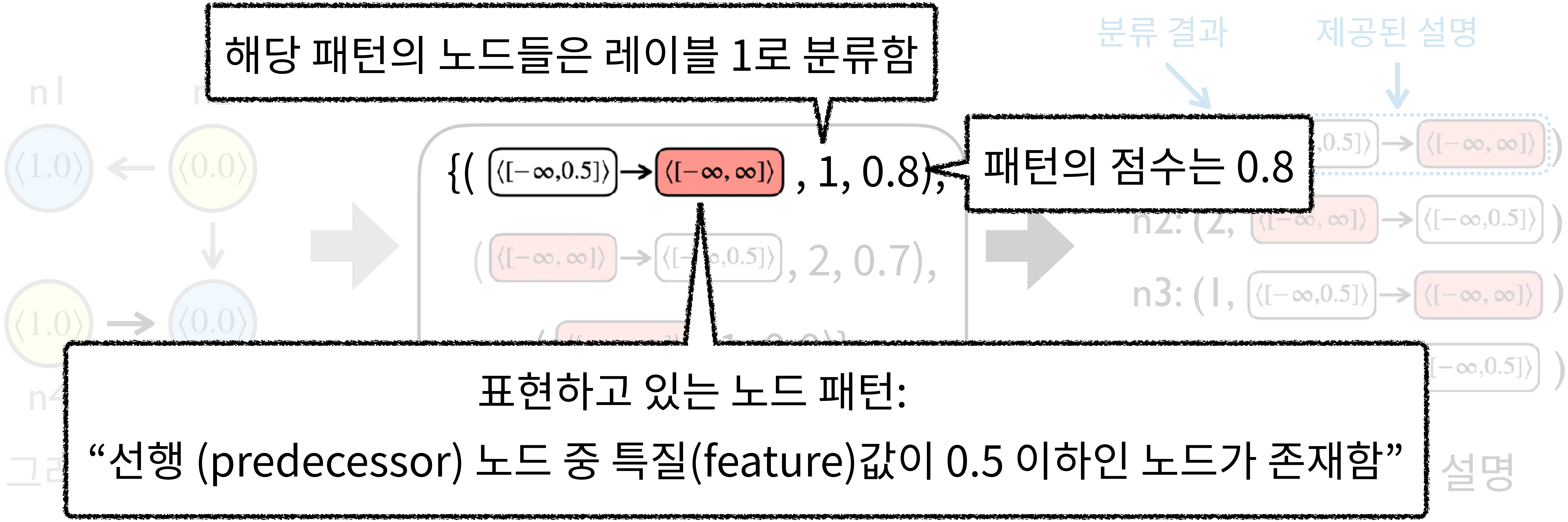
Programs	P	$::= \overline{\delta}$ **target** t	$\in$	$\mathbb{P}$	$= \mathbb{D}^* \times \mathbb{T}$
Descriptions	δ	$::= \delta_V \mid \delta_E$	$\in$	$\mathbb{D}$	$= \mathbb{D}_V \uplus \mathbb{D}_E$
Node Descriptions	δ_V	$::=$ **node** $x <\overline{\phi}>^?$	$\in$	$\mathbb{D}_V$	$= \mathbb{X} \times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

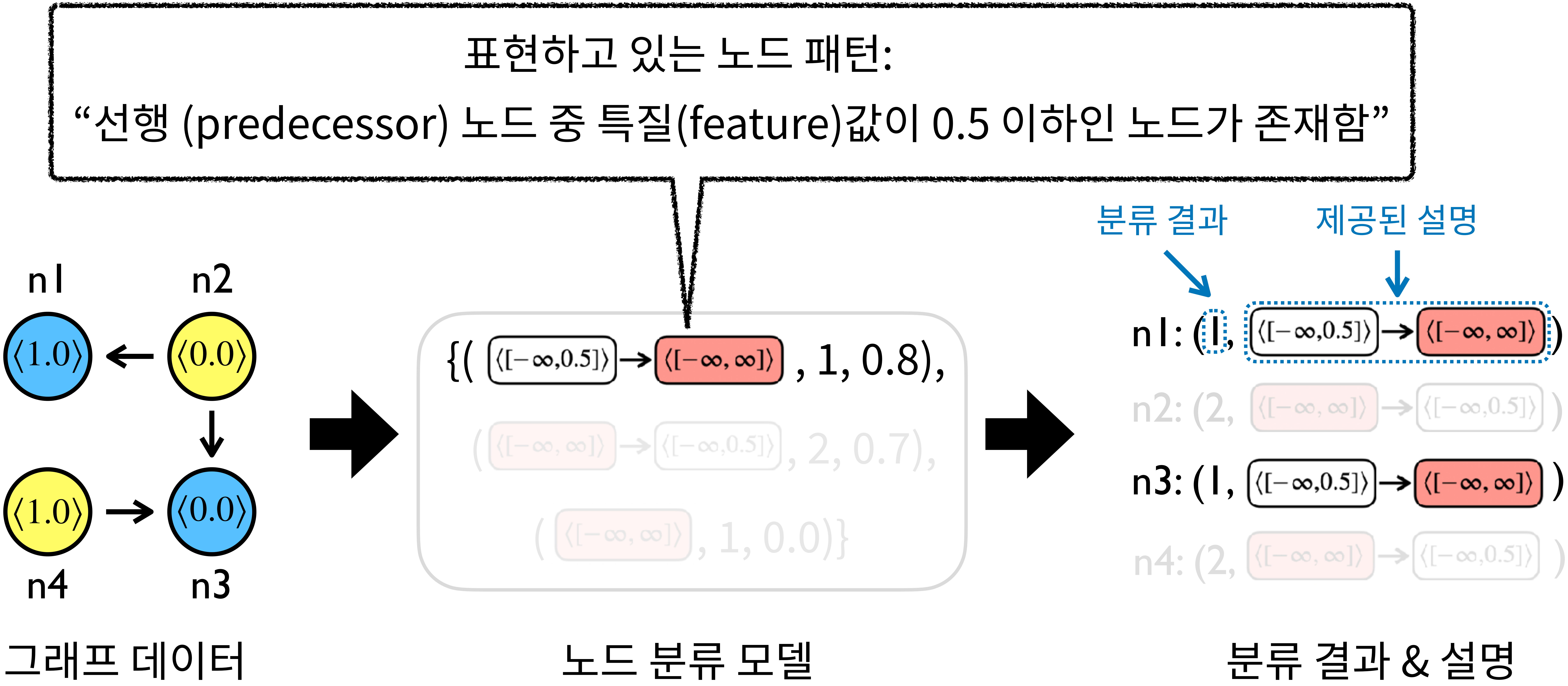
node x $\langle[-\infty, \infty]\rangle$
node y $\langle[-\infty, 0.5]\rangle$
edge (x, y)
target node x



표현하고 있는 노드 패턴:

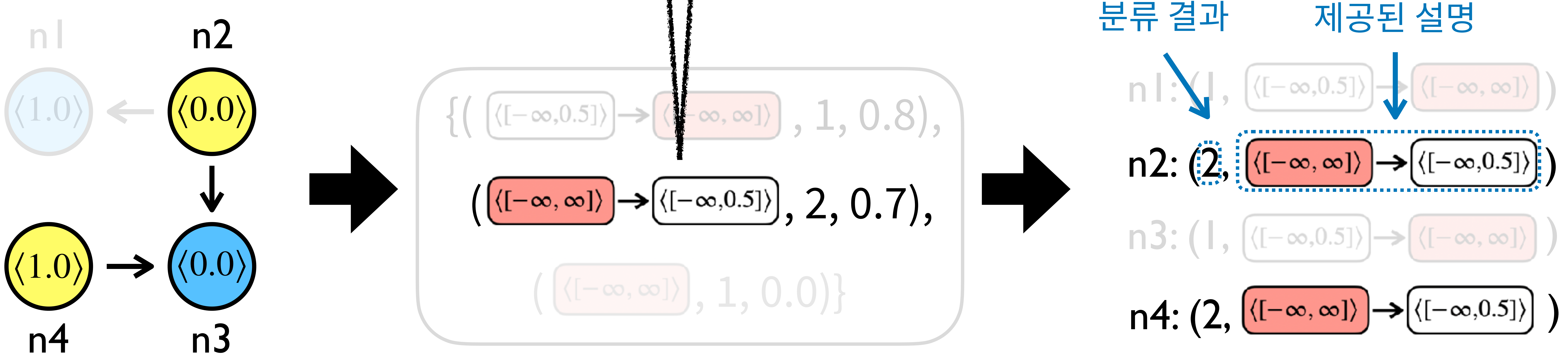
“선행 (predecessor) 노드 중 특질(feature)값이 0.5 이하인 노드가 존재함”


해당 패턴의 노드들은 레이블 1로 분류함
분류 결과
제공된 설명
{(⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩, 2, 0.7),
패턴의 점수는 0.8
n2: (2,
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
표현하고 있는 노드 패턴:
“선행 (predecessor) 노드 중 특질(feature)값이 0.5 이하인 노드가 존재함”
설명


표현하고 있는 노드 패턴:
“선행 (predecessor) 노드 중 특질(feature)값이 0.5 이하인 노드가 존재함”
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)}
분류 결과
제공된 설명
n1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
그래프 데이터
노드 분류 모델
분류 결과 & 설명

표현하고 있는 노드 패턴:

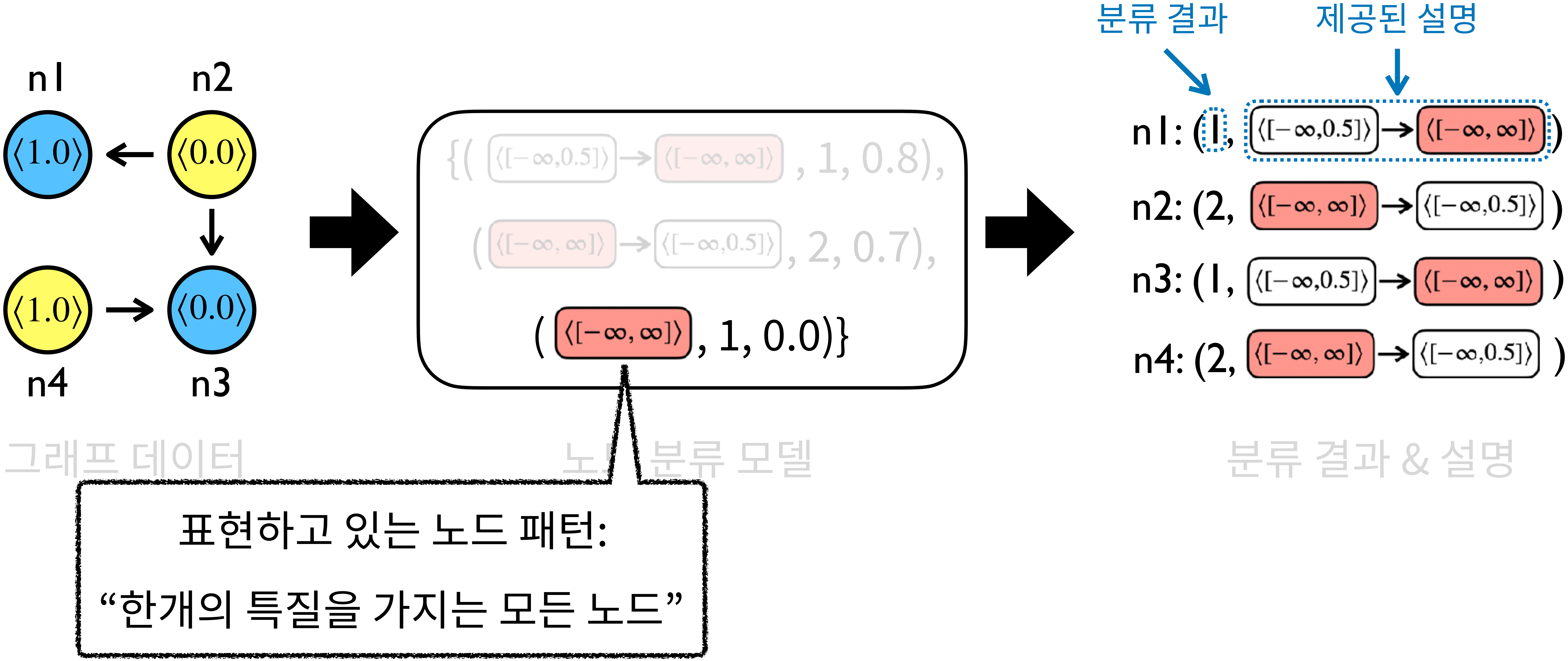
"후속 (successor) 노드 중 특질(feature)값이 0.5 이하인 노드가 존재함"



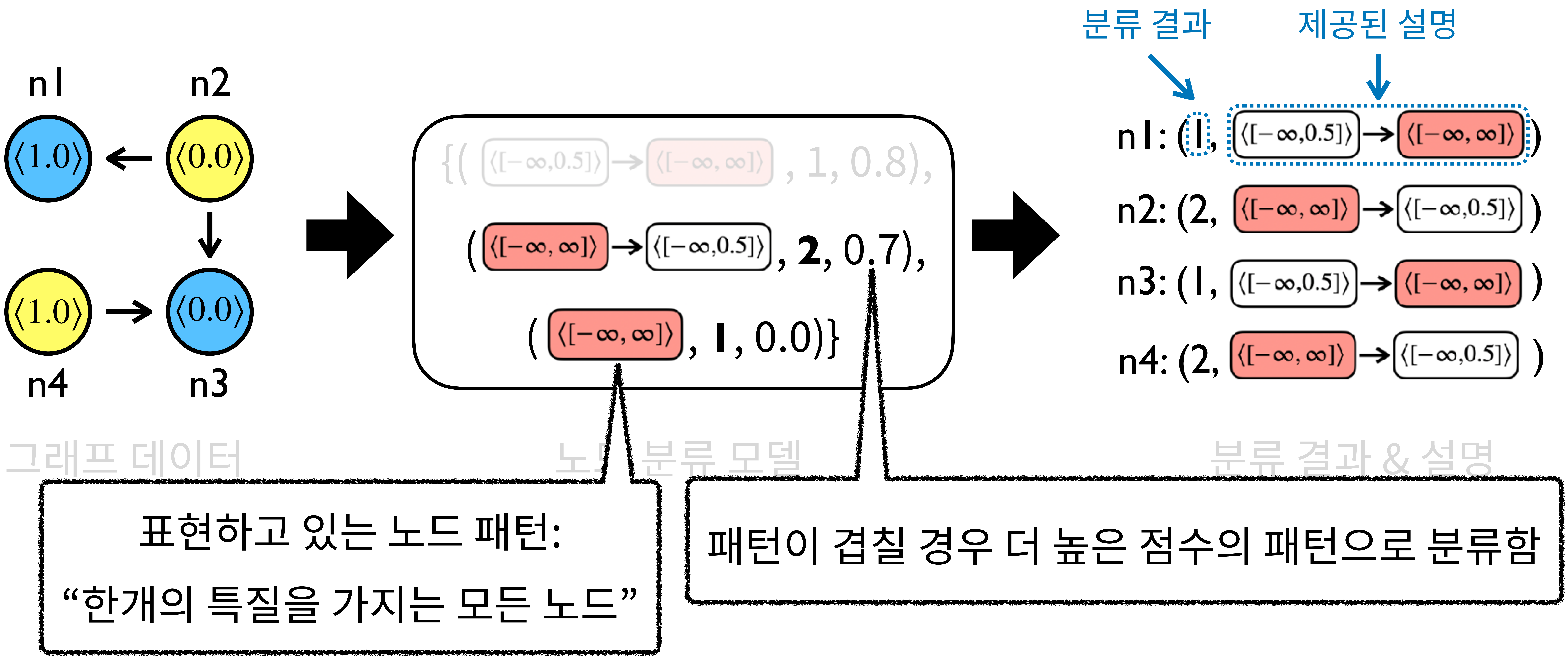
그래프 데이터

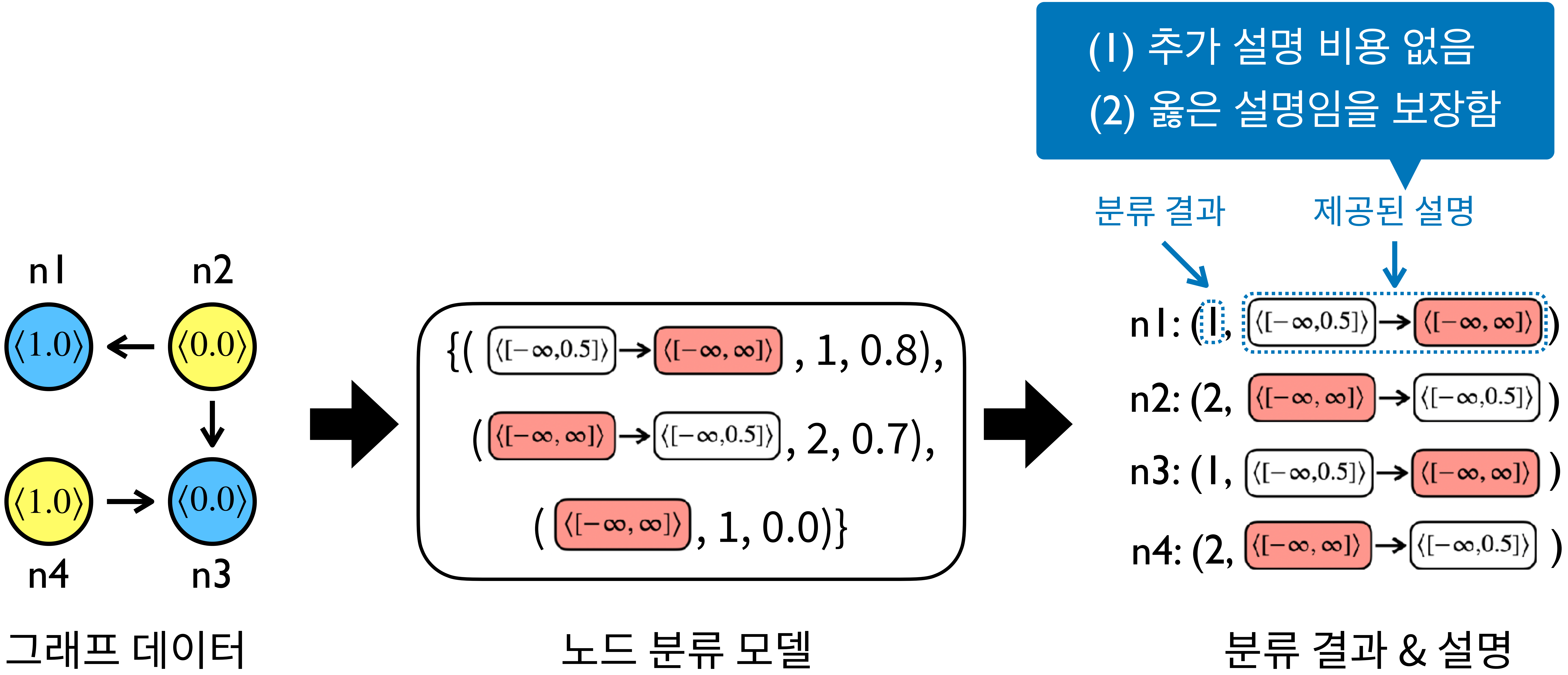
노드 분류 모델

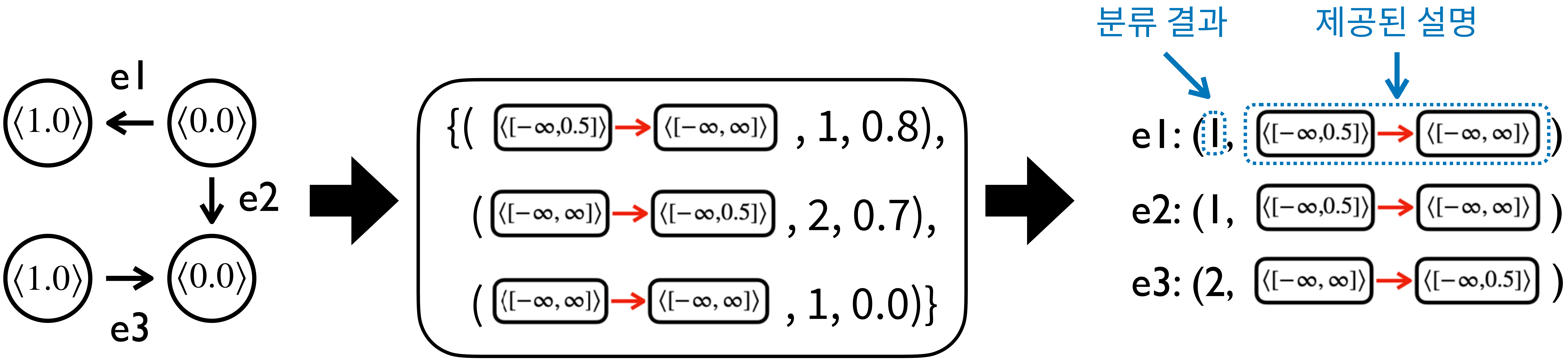
분류 결과 & 설명



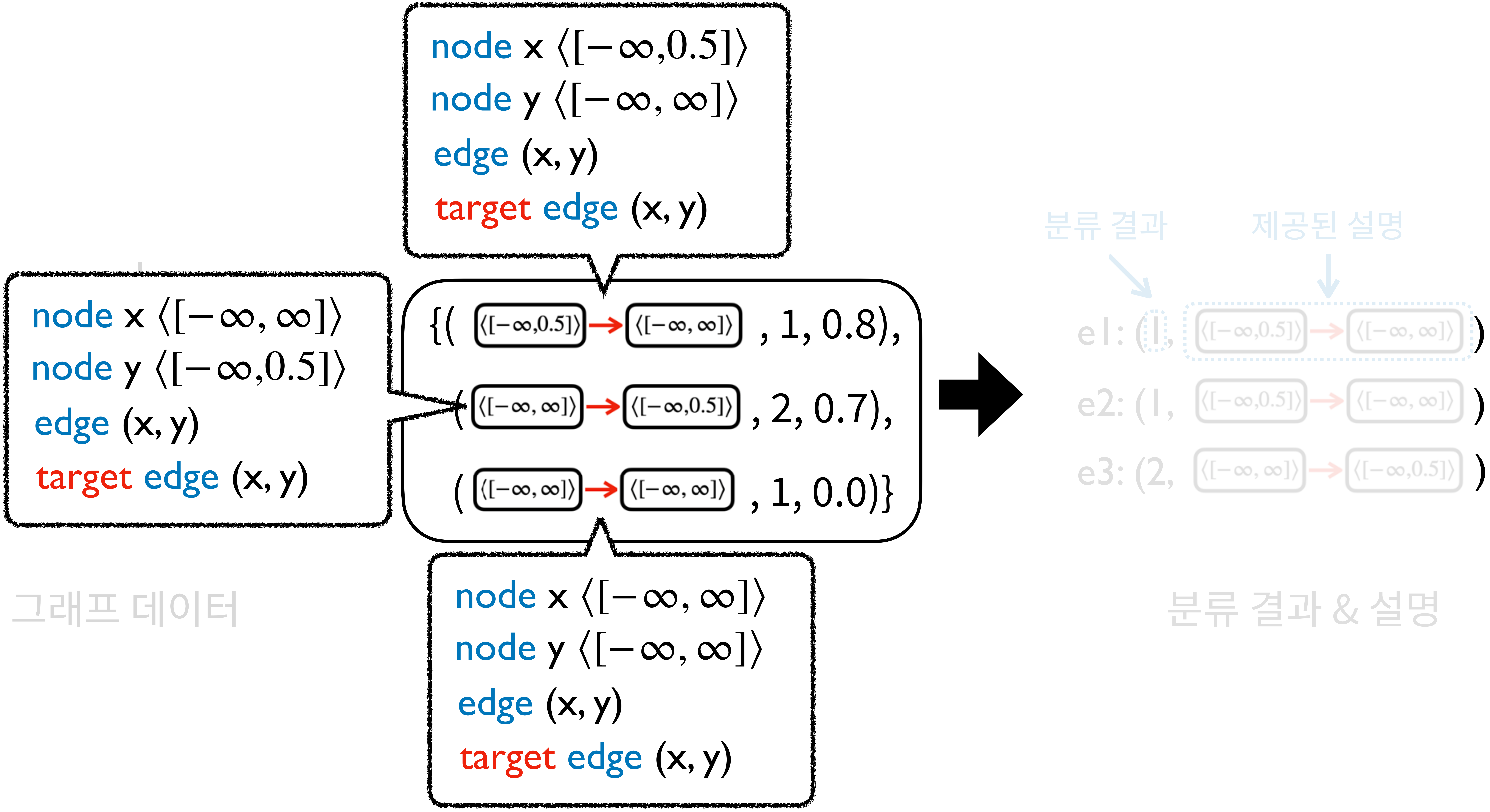
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
그래프 데이터
{(⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩, 1, 0.8),
(⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
노드 분류 모델
분류 결과
제공된 설명
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
분류 결과 & 설명
표현하고 있는 노드 패턴:
“한개의 특질을 가지는 모든 노드”

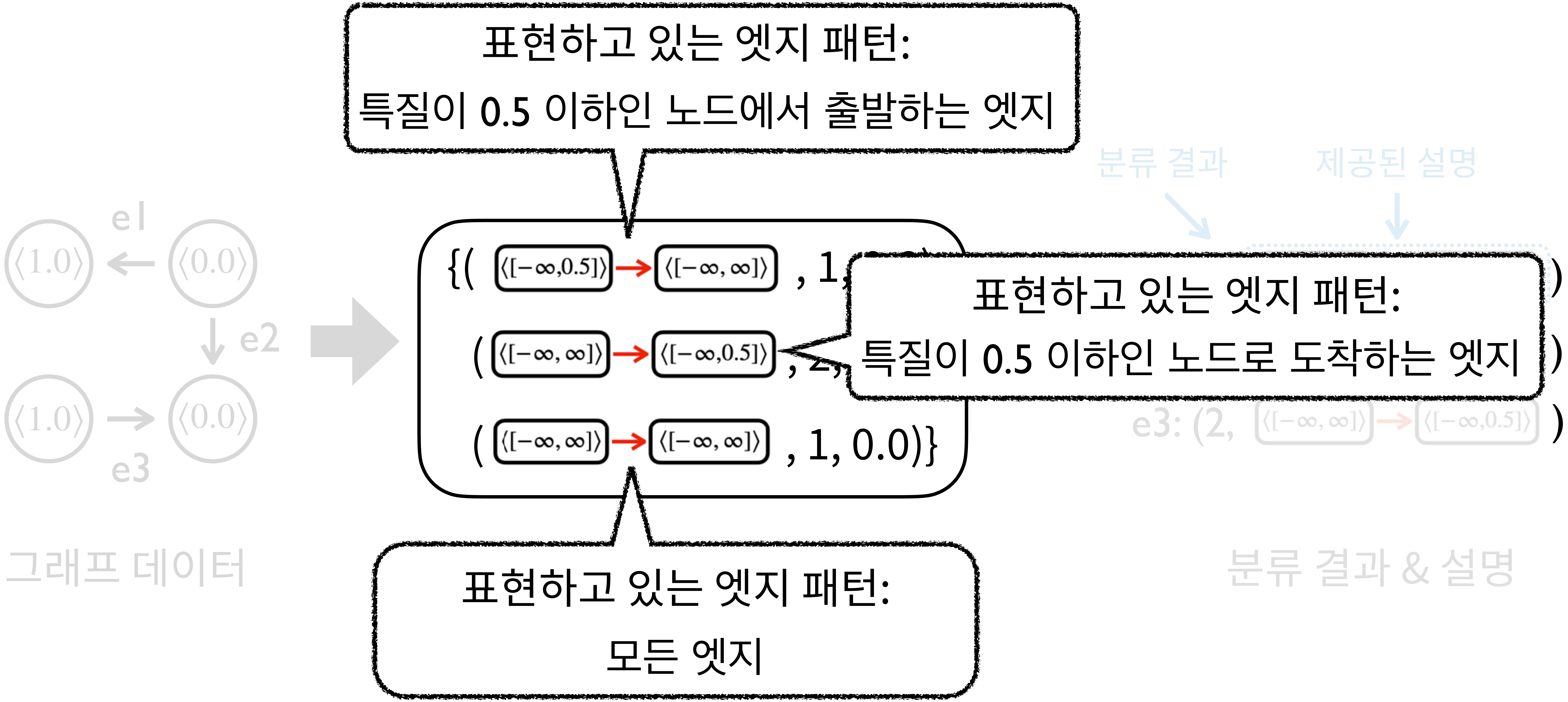

n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
그래프 데이터
{(⟨[−∞,0.5]⟩→⟨[−∞,∞]⟩, 1, 0.8),
(⟨[−∞,∞]⟩→⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞,∞]⟩, 1, 0.0)}
분류 모델
분류 결과
제공된 설명
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞,∞]⟩)
n2: (2, ⟨[−∞,∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞,∞]⟩)
n4: (2, ⟨[−∞,∞]⟩→⟨[−∞,0.5]⟩)
분류 결과 & 설명
표현하고 있는 노드 패턴:
“한개의 특질을 가지는 모든 노드”
패턴이 겹칠 경우 더 높은 점수의 패턴으로 분류함


(1) 추가 설명 비용 없음
(2) 옳은 설명임을 보장함
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)}
분류 결과
제공된 설명
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
그래프 데이터
노드 분류 모델
분류 결과 & 설명

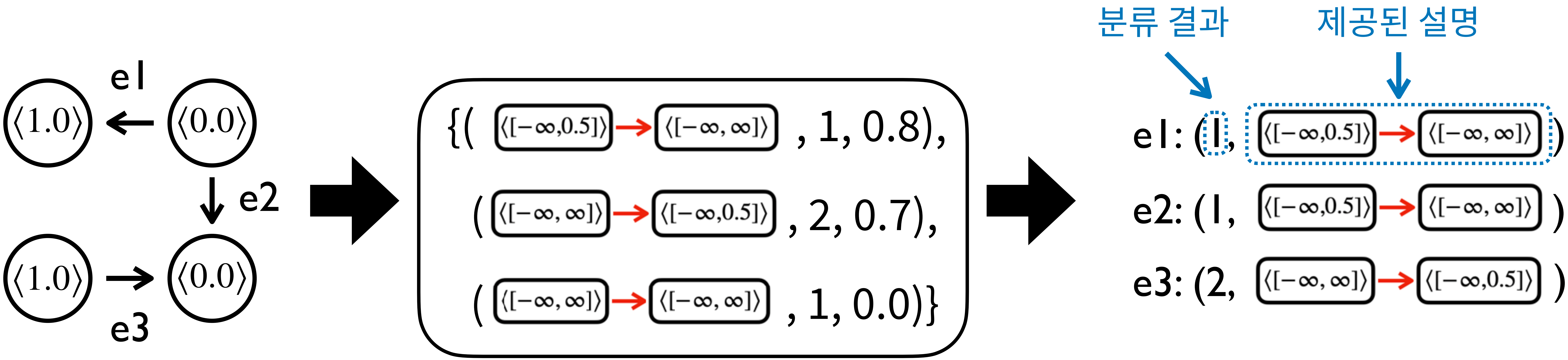


그래프 데이터　　엣지 분류 모델　　분류 결과 & 설명

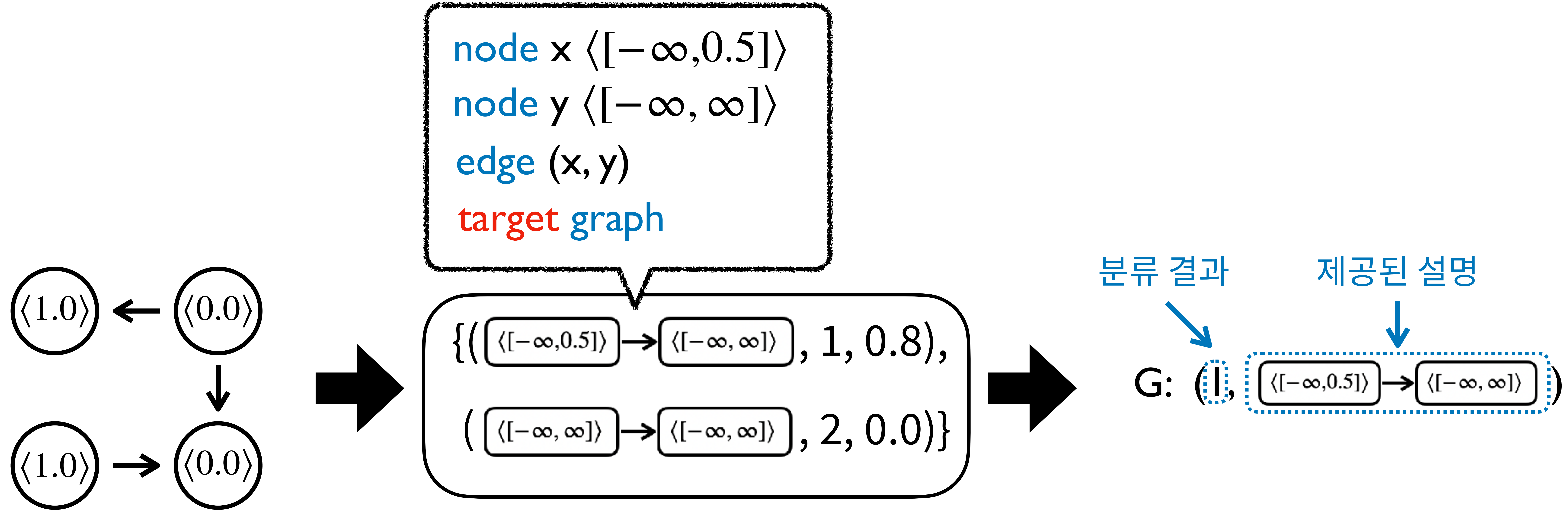

node x ⟨[−∞,0.5]⟩
node y ⟨[−∞, ∞]⟩
edge (x, y)
target edge (x, y)
node x ⟨[−∞, ∞]⟩
node y ⟨[−∞,0.5]⟩
edge (x, y)
target edge (x, y)
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩ , 2, 0.7),
(⟨[−∞, ∞]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.0)}
node x ⟨[−∞, ∞]⟩
node y ⟨[−∞, ∞]⟩
edge (x, y)
target edge (x, y)
그래프 데이터
분류 결과
제공된 설명
e1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
e2: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
e3: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
분류 결과 & 설명



표현하고 있는 엣지 패턴:
특질이 0.5 이하인 노드에서 출발하는 엣지
분류 결과
제공된 설명
e1
⟨1.0⟩
⟨0.0⟩
e2
⟨1.0⟩
⟨0.0⟩
e3
그래프 데이터
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1,
(⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩ , 2,
(⟨[−∞, ∞]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.0)}
표현하고 있는 엣지 패턴:
특질이 0.5 이하인 노드로 도착하는 엣지
e3: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
분류 결과 & 설명
표현하고 있는 엣지 패턴:
모든 엣지



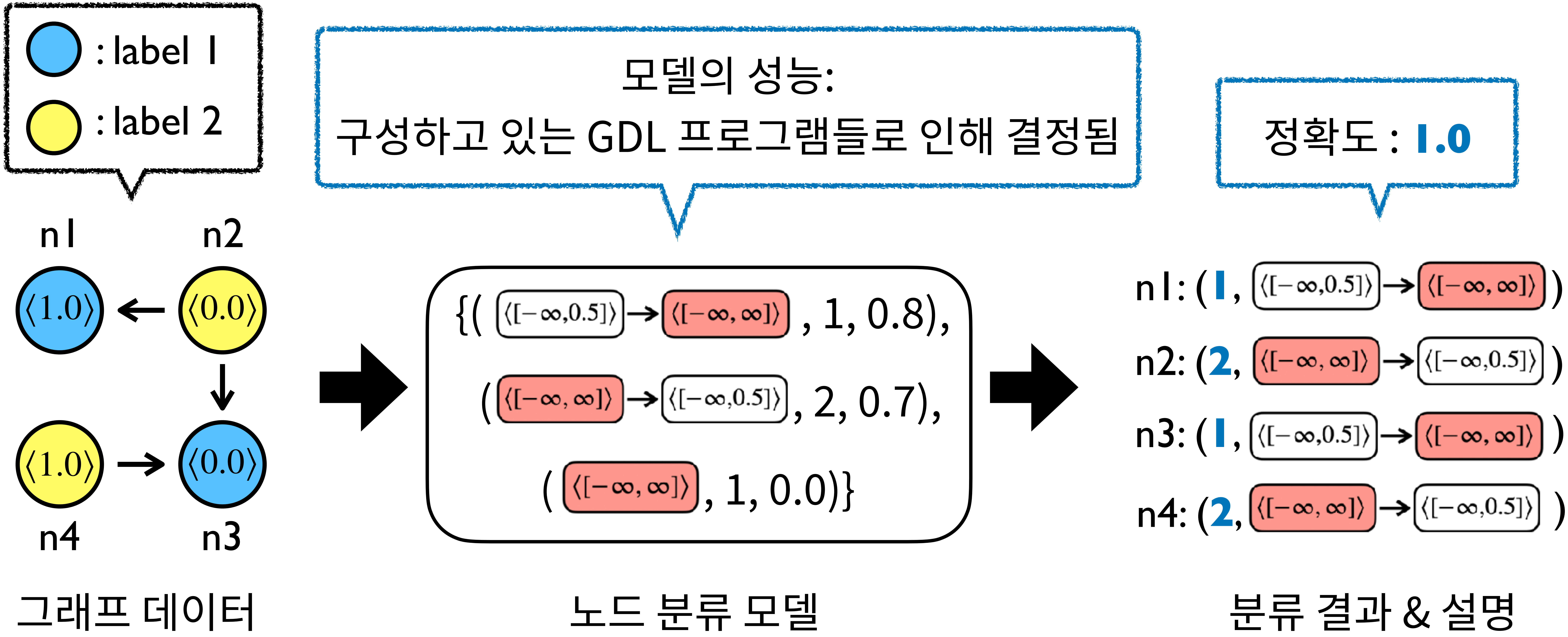
그래프 데이터 엣지 분류 모델 분류 결과 & 설명

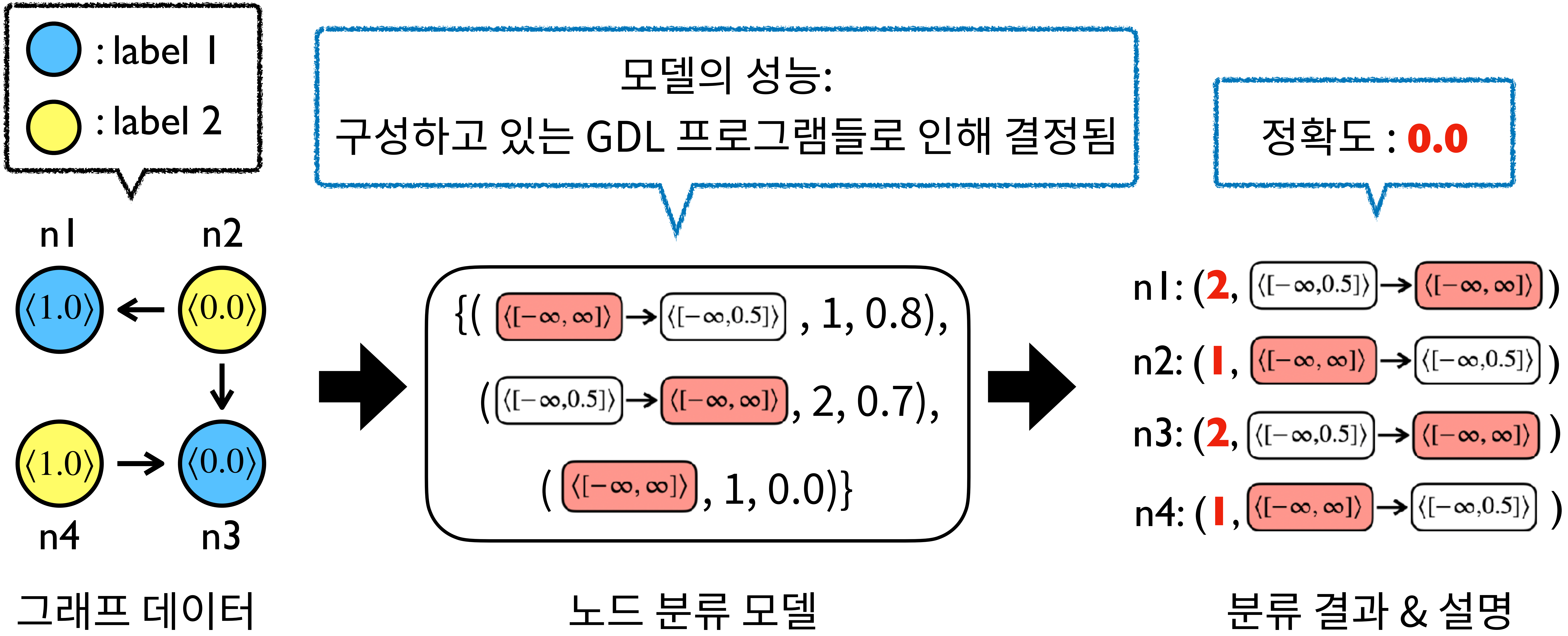


그래프 데이터 G

그래프 분류 모델

분류 결과 & 설명


: label 1
: label 2
n1
n2
n3
n4
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
모델의 성능:
구성하고 있는 GDL 프로그램들로 인해 결정됨
{(⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩, 1, 0.8),
(⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
정확도 : 1.0
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
그래프 데이터
노드 분류 모델
분류 결과 & 설명



: label 1
: label 2
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
그래프 데이터
모델의 성능:
구성하고 있는 GDL 프로그램들로 인해 결정됨
{(⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩ , 1, 0.8),
(⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
노드 분류 모델
정확도 : 0.0
n1: (2, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n2: (1, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
n3: (2, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n4: (1, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
분류 결과 & 설명

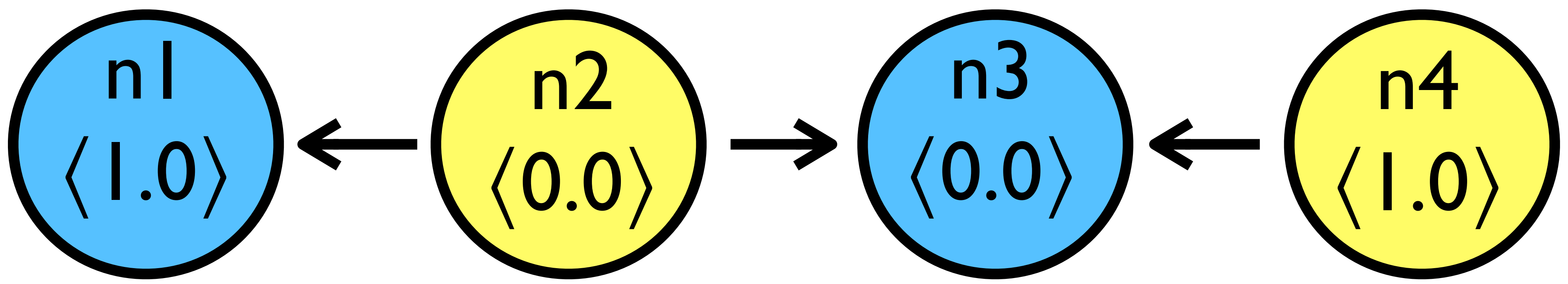
학습 데이터
(학습 그래프 데이터)
GDL 프로그램
합성 알고리즘
학습 목표:
고품질 GDL 프로그램 합성하기
n1
n2
n4
n3
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
n1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
그래프 데이터
노드 분류 모델
분류 결과 & 설명

- 임의의 그래프에 대해서 더 많이 고르는 프로그램일 수록 더 큰 (일반적인) 프로그램

GDL 프로그램 합성 알고리즘

타겟 노드 = n1

학습 데이터(그래프) :



: label 1

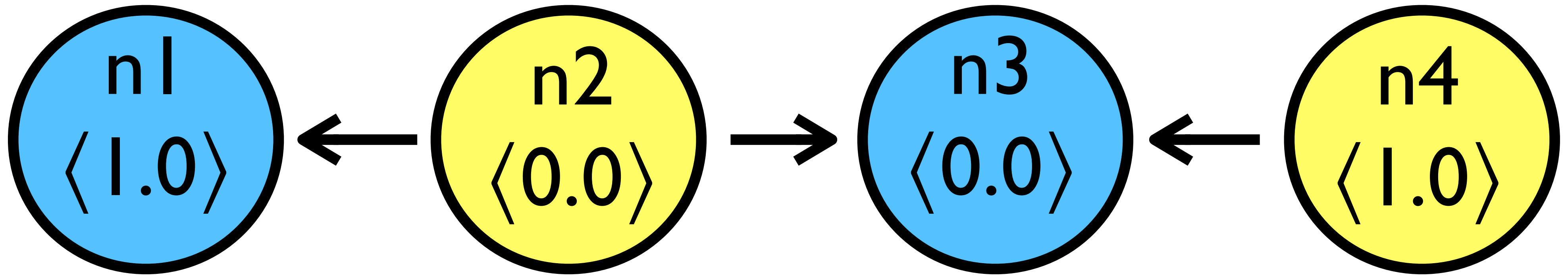
: label 2

학습 목표

n1(타겟 노드)를 포함하면서, label 1()에 해당하는 노드들을 정밀(precise)하게 표현하는 GDL프로그램 합성하기

세분화(Top-down) GDL 프로그램 합성 알고리즘

학습 데이터(그래프) :



타겟 노드 = n1

: label 1

: label 2

(1) 가장 일반적인 패턴(간단한 GDL 프로그램)에서 부터 시작

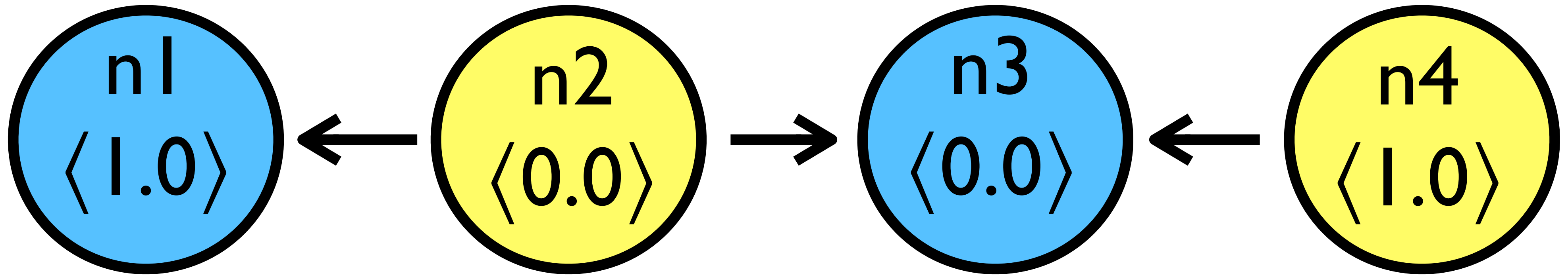
$\langle[-\infty, \infty]\rangle$

Score : 0.4

$$\frac{|\{\mathbf{n1}, n3\}|}{|\{\mathbf{n1}, n2, n3, n4\}| + 1}$$

세분화(Top-down) GDL 프로그램 합성 알고리즘

학습 데이터(그래프) :

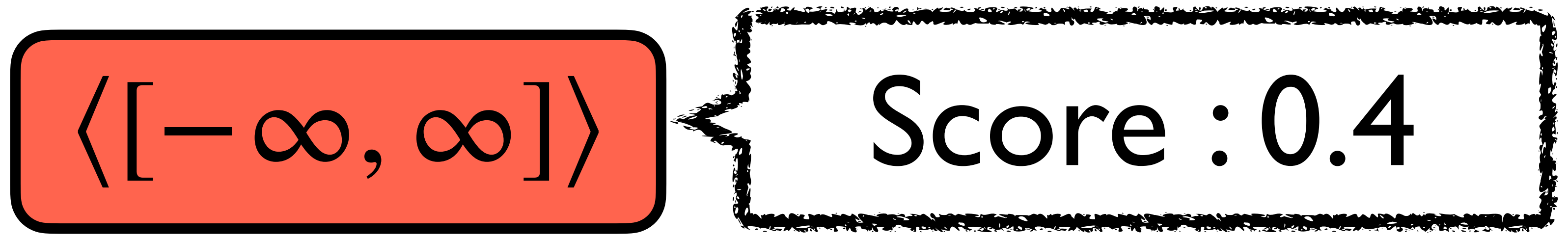


타겟 노드 = n1

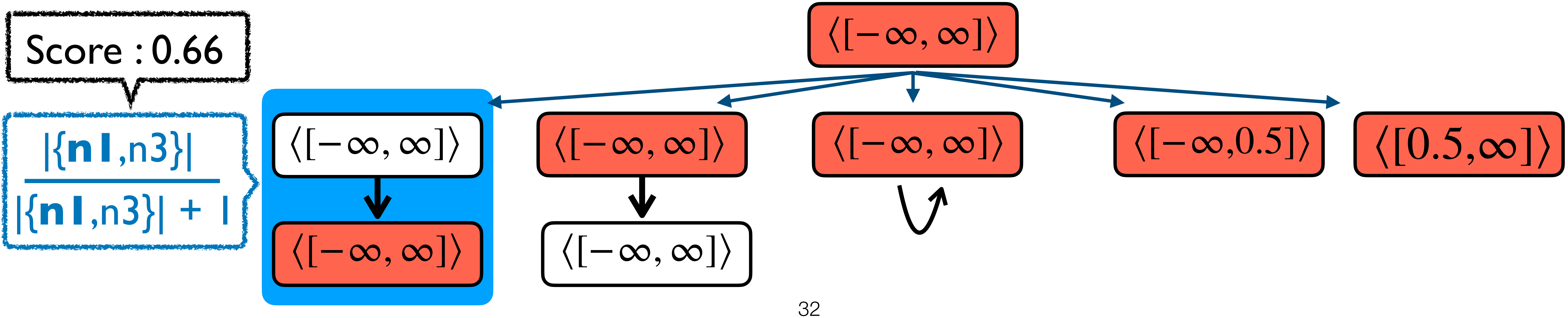
: label 1

: label 2

(1) 가장 일반적인 패턴(간단한 GDL 프로그램)에서 부터 시작



(2) 주어진 패턴을 다양하게 세분화하여 나열 후 더 높은 점수의 프로그램 선택 (나열 탐색)



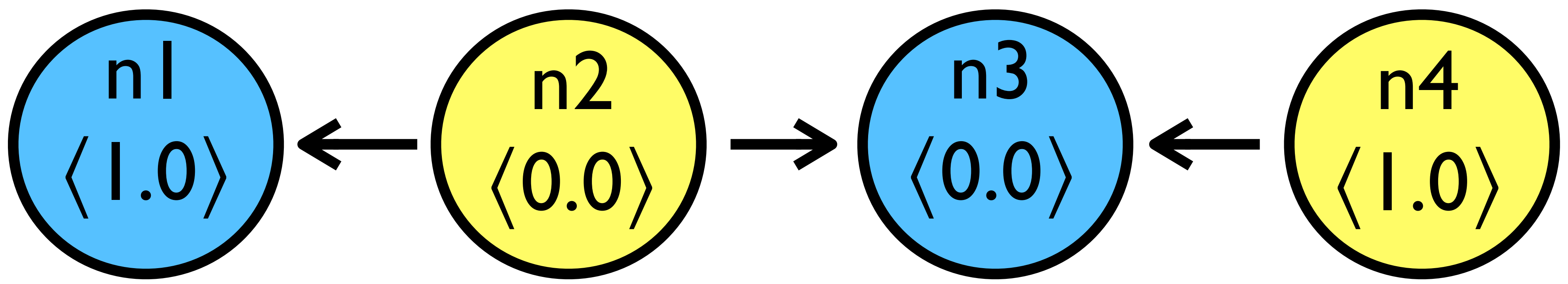
세분화(Top-down) GDL 프로그램 합성 알고리즘

타겟 노드 = n1

: label 1

: label 2

학습 데이터(그래프) :



(1) 가장 일반적인 패턴(간단한 GDL 프로그램)에서 부터 시작

$\langle[-\infty, \infty]\rangle$ Score : 0.4

(2) 주어진 패턴을 다양하게 세분화하여 나열 후 더 높은 점수의 프로그램 선택 (나열 탐색)

$\langle[-\infty, \infty]\rangle \rightarrow \langle[-\infty, \infty]\rangle$ Score : 0.66

(3) 모든 나열된 패턴이 현재 패턴보다 같거나 낮은 점수를 가질 때 까지 (2)를 반복

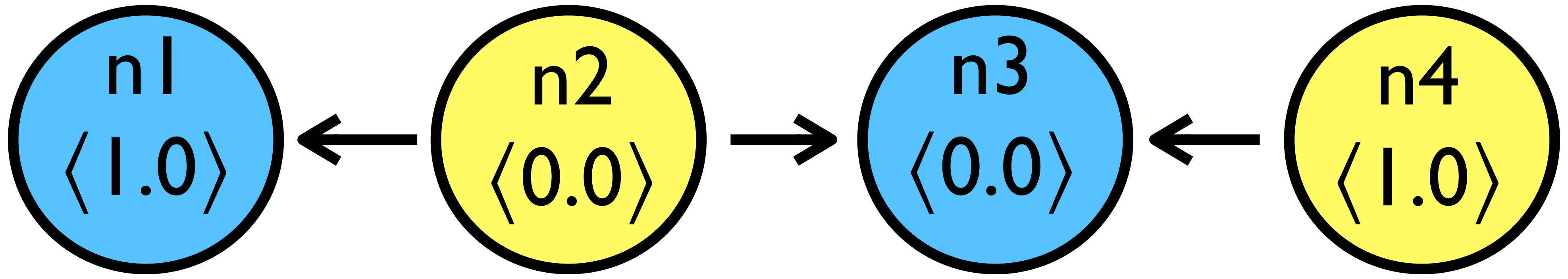
세분화(Top-down) GDL 프로그램 합성 알고리즘

타겟 노드 = n1

: label 1

: label 2

학습 데이터(그래프) :



(1) 가장 일반적인 패턴(간단한 GDL 프로그램)에서 부터 시작

$\langle[-\infty, \infty]\rangle$ Score : 0.4

(2) 주어진 패턴을 다양하게 세분화하여 나열 후 더 높은 점수의 프로그램 선택 (나열 탐색)

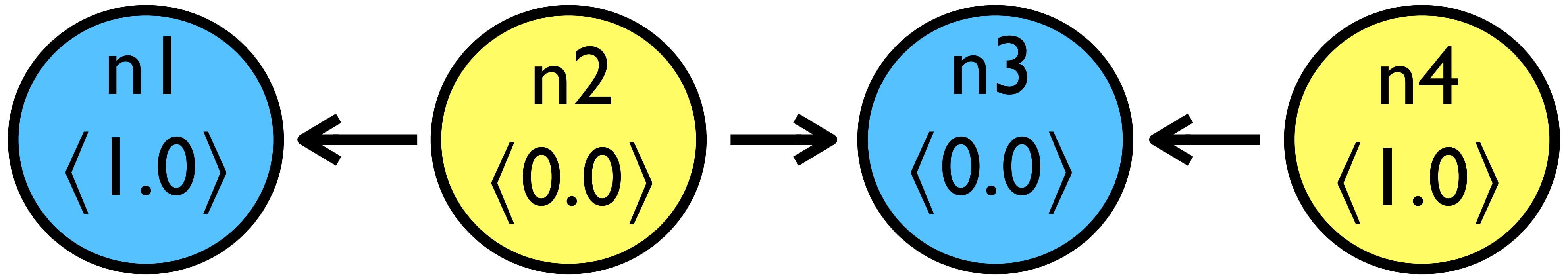
$\langle[-\infty, \infty]\rangle \rightarrow \langle[-\infty, \infty]\rangle$ Score : 0.66

(3) 모든 나열된 패턴이 현재 패턴보다 같거나 낮은 점수를 가질 때 까지 (2)를 반복

(4) 현재 패턴을 반환 $(\langle[-\infty, \infty]\rangle \rightarrow \langle[-\infty, \infty]\rangle$, label 1, 0.66)

일반화(Bottom-up) GDL 프로그램 합성 알고리즘

학습 데이터(그래프) :

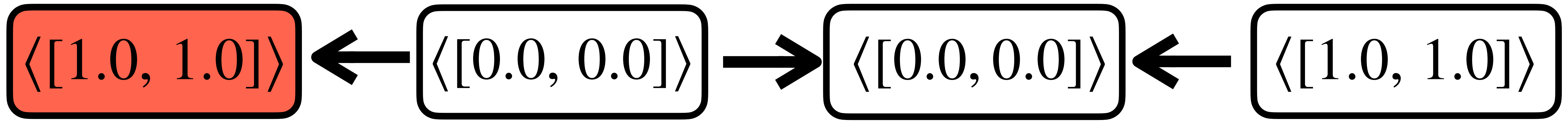


타겟 노드 = n1

: label 1

: label 2

(1) 가장 세분화한 패턴(복잡한 GDL 프로그램)부터 시작



Score : 0.5

$$\frac{|\{n1\}|}{|\{n1\}| + 1}$$

```
node v1 ⟨[1.0,1.0]⟩
node v2 ⟨[0.0,0.0]⟩
node v3 ⟨[0.0,0.0]⟩
node v4 ⟨[0.0,0.0]⟩
edge (v2, v1)
edge (v2, v3)
edge (v4, v3)
target node v1
```

일반화(Bottom-up) GDL 프로그램 합성 알고리즘

타겟 노드 = n1

학습 데이터(그래프) : n1 ⟨1.0⟩ ← n2 ⟨0.0⟩ → n3 ⟨0.0⟩ ← n4 ⟨1.0⟩

: label 1

: label 2

(1) 가장 세분화한 패턴(복잡한 GDL 프로그램)부터 시작

⟨[1.0, 1.0]⟩ ← ⟨[0.0, 0.0]⟩ → ⟨[0.0, 0.0]⟩ ← ⟨[1.0, 1.0]⟩ Score : 0.5

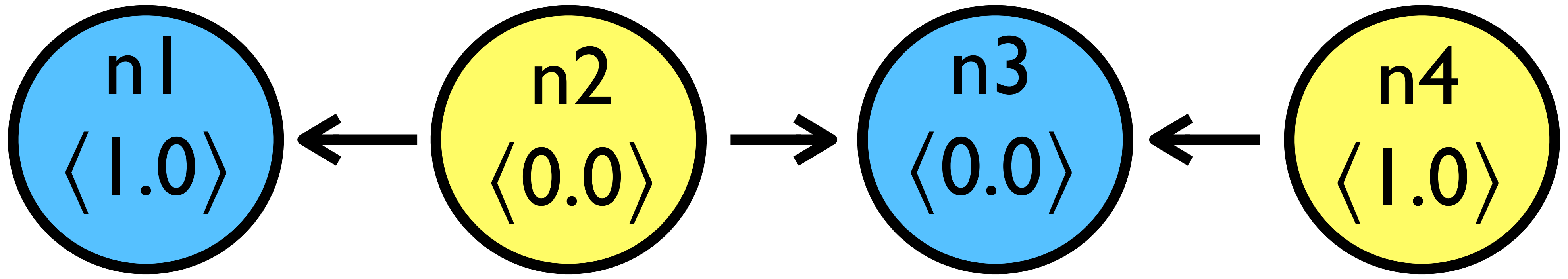
(2) 주어진 패턴을 다양하게 간단화하여 나열 후 같거나 더 높은 점수의 패턴을 선택 (나열 탐색)

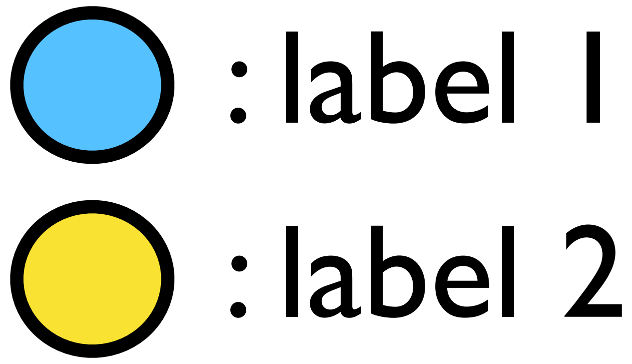
⟨[1.0, 1.0]⟩ ← ⟨[0.0, 0.0]⟩ → ⟨[0.0, 0.0]⟩ Score : 0.5

일반화(Bottom-up) GDL 프로그램 합성 알고리즘

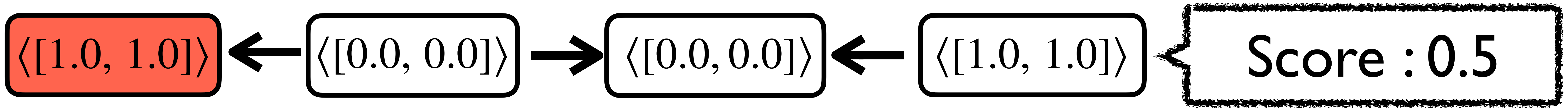
타겟 노드 = n1

학습 데이터(그래프) :

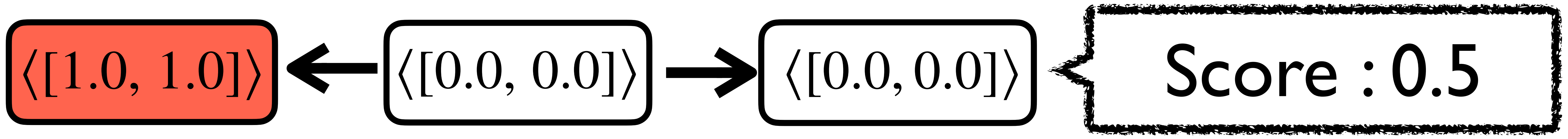




(1) 가장 세분화한 패턴(복잡한 GDL 프로그램)부터 시작



(2) 주어진 패턴을 다양하게 간단화하여 나열 후 같거나 더 높은 점수의 패턴을 선택 (나열 탐색)



(3) 모든 나열된 패턴이 현재 패턴보다 낮은 점수를 가질 때 까지 (2)를 반복

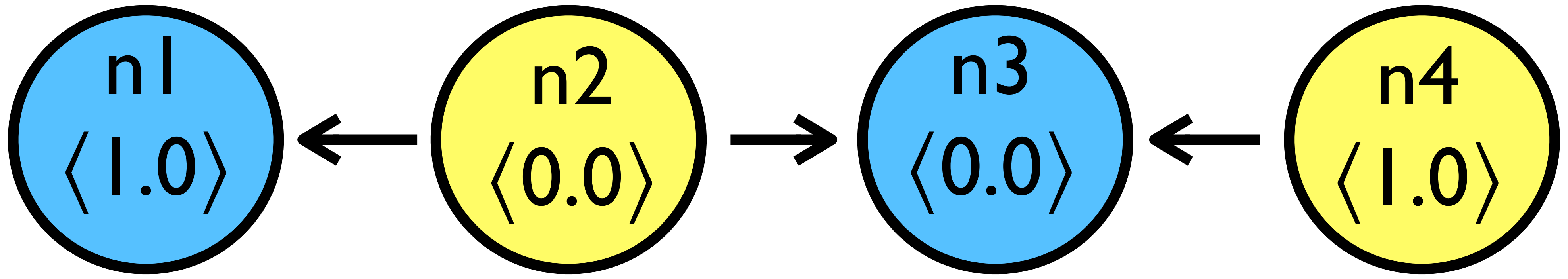
일반화(Bottom-up) GDL 프로그램 합성 알고리즘

타겟 노드 = n1

: label 1

: label 2

학습 데이터(그래프) :



(1) 가장 세분화한 패턴(복잡한 GDL 프로그램)부터 시작

⟨[1.0, 1.0]⟩ ← ⟨[0.0, 0.0]⟩ → ⟨[0.0, 0.0]⟩ ← ⟨[1.0, 1.0]⟩ Score : 0.5

(2) 주어진 패턴을 다양하게 간단화하여 나열 후 같거나 더 높은 점수의 패턴을 선택 (나열 탐색)

⟨[1.0, 1.0]⟩ ← ⟨[0.0, 0.0]⟩ → ⟨[0.0, 0.0]⟩ Score : 0.5

(3) 모든 나열된 패턴이 현재 패턴보다 낮은 점수를 가질 때 까지 (2)를 반복

(4) 현재 패턴을 반환 ($\langle[-\infty, \infty]\rangle \leftarrow \langle[-\infty, \infty]\rangle$, 1, 0.66)


타겟 노드 = n1
n1 ⟨1.0⟩
cpu
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 1, 0.66)
타겟 노드 = n2
n2 ⟨0.0⟩
cpu
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 2, 0.66)
타겟 노드 = n3
n3 ⟨0.0⟩
cpu
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 1, 0.66)
타겟 노드 = n4
n4 ⟨1.0⟩
cpu
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 2, 0.66)
학습 데이터
{(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 1, 0.66),
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 2, 0.66)}
학습된 노드 분류 모델

타겟 노드 = n1
n1 ⟨1.0⟩
cpu
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 1, 0.66)
타겟 노드 = n2
n2 ⟨0.0⟩
cpu
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 2, 0.66)
타겟 노드 = n3
n3 ⟨0.0⟩
cpu
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 1, 0.66)
타겟 노드 = n4
n4 ⟨1.0⟩
cpu
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 2, 0.66)
학습 데이터
{(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 1, 0.66),
(⟨[−∞, ∞]⟩ ← ⟨[−∞, ∞]⟩, 2, 0.66),
(⟨[−∞, ∞]⟩, 1, 0.0) }
학습된 노드 분류 모델
기본값 추가

실험

- RQ 1 (**정확도**): 우리의 방법(PL4XGL)은 정확하게 분류하는가?
- RQ 2 (**비용**): 우리의 방법(PL4XGL)의 (학습/분류/설명) 비용은 어떠한가?
- RQ 3 (**설명력**): 우리의 방법(PL4XGL)의 설명의 품질은 어떠한가?

(RQ 1) 정확도 비교

	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

그래프 (분자 구조) 분류 데이터 셋

MUTAG 데이터 셋

- 분자구조 그래프 분류 데이터 셋
  - 총 188개의 그래프 (총 3771개의 노드)
  - 각 노드는 7개중 하나의 특질을 가짐

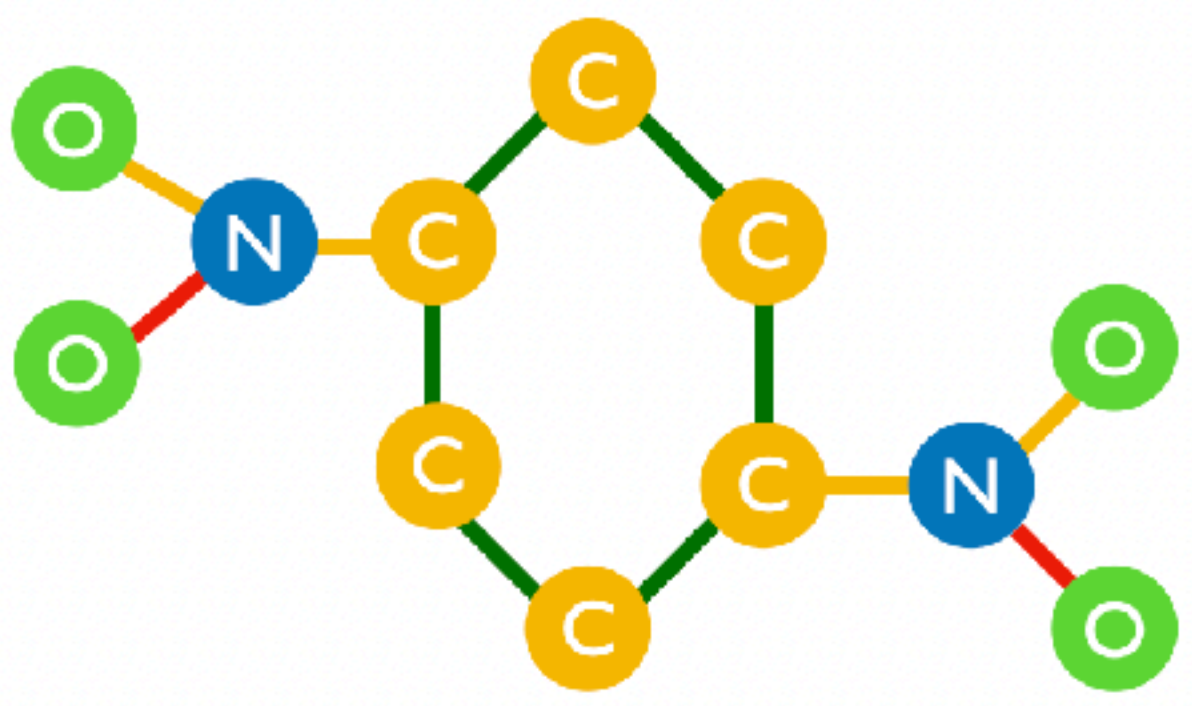
C N O F I Cl Br

  - 각 엣지는 4개중 하나의 특질을 가짐

aromatic single double triple

  - 분자의 박테리아 반응 여부 (양성/음성)

MUTAG에 있는 그래프



							GCN	PL4XGL
MUTAG							N/A	100.0±0.0
BBBP	83						N/A	86.8±0.0
BACE	78						N/A	80.9±0.0
HIV	96						N/A	N/A
BA-SHAPES	95						.1±0.7	95.7±0.0
TREE-CYCLES	97						.2±0.5	100.0±0.0
WISCONSIN	64						.0±0.0	88.0±0.0
TEXAS	67						.6±2.6	83.3±0.0
CORNELL	58						.6±2.6	88.8±0.0
CORA	85						.2±5.9	80.0± 0.0
CITESEER	75.2±0.0	74.5±0.7	79.1±0.9	75.7±4.2	75.9±2.5	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

(RQ 1) 정확도 비교

	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-								
TREE								
WIS								
T								
CO								
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

분자구조 분류 데이터셋에서 GNN보다 높은 정확도를 보임

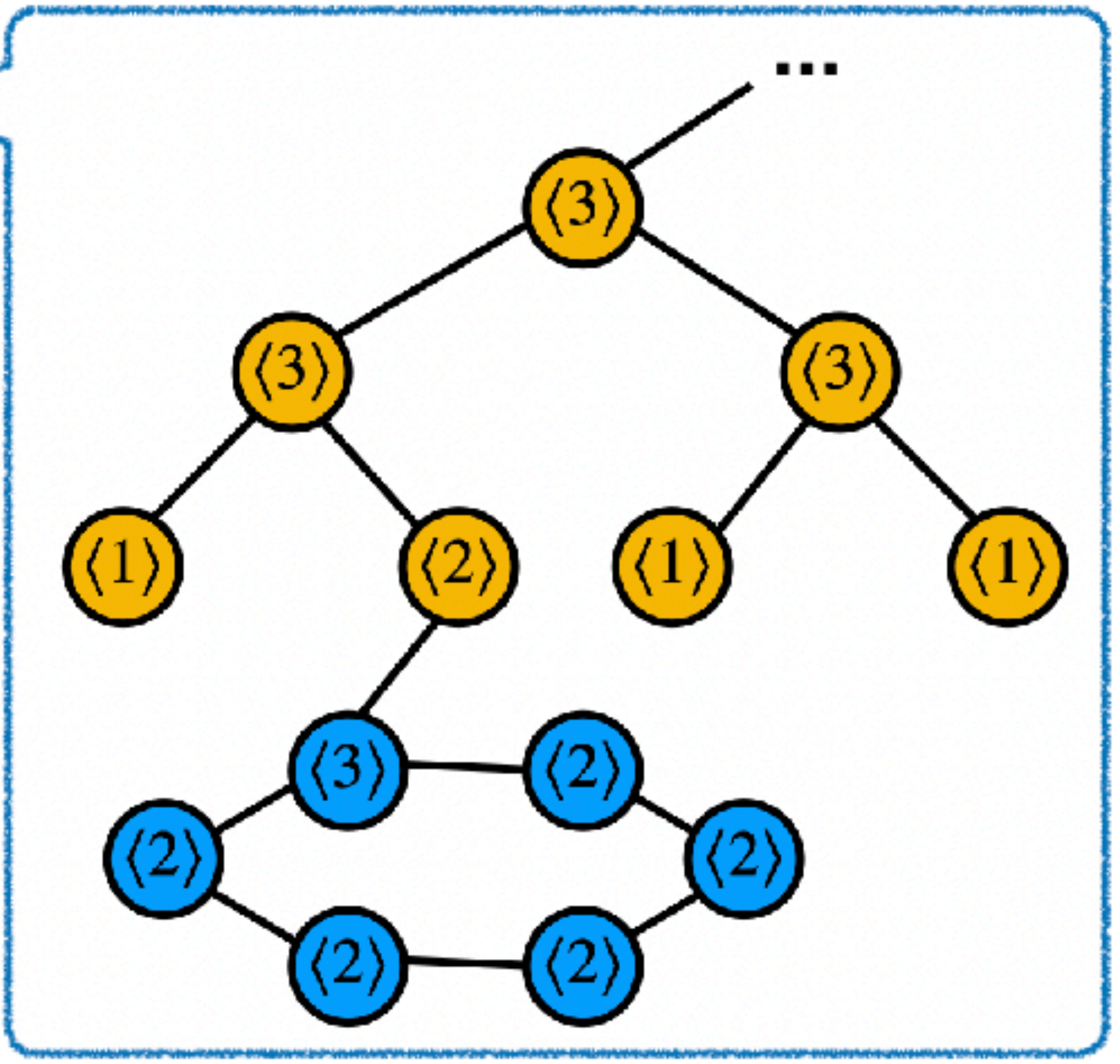
- 분자구조 데이터셋은 신약 개발과 관련되어 분류에 대한 설명이 요구됨

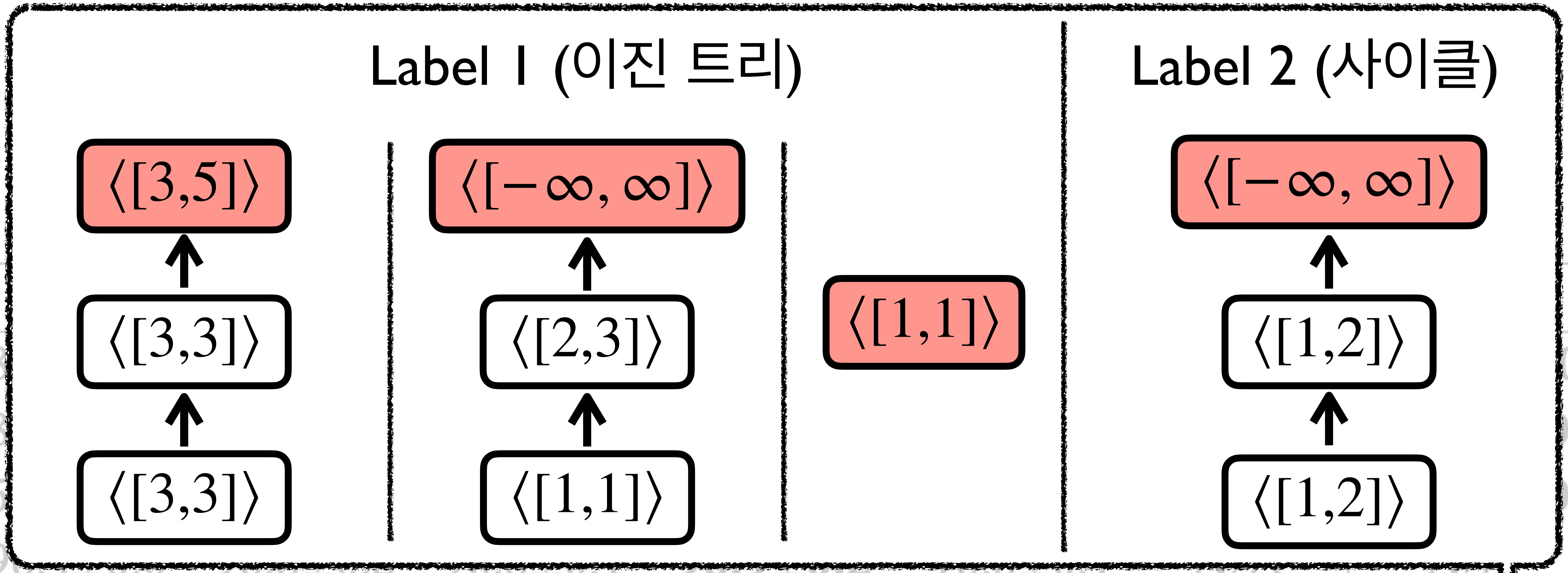
(RQ 1) 정확도 비교

	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						A	**80.9±0.0**
HIV	96						A	N/A
BA-Shapes	95						±0.7	95.7±0.0
Tree-Cycles							±0.5	**100.0±0.0**
Wisconsin	64						**±0.0**	88.0±0.0
Texas	67						**±2.6**	83.3±0.0
Cornell	58						±2.6	**88.8±0.0**
Cora	85						±5.9	80.0± 0.0
Citeseer	75						±6.0	63.8± 0.0
Pubmed	82						±0.6	81.4±0.0

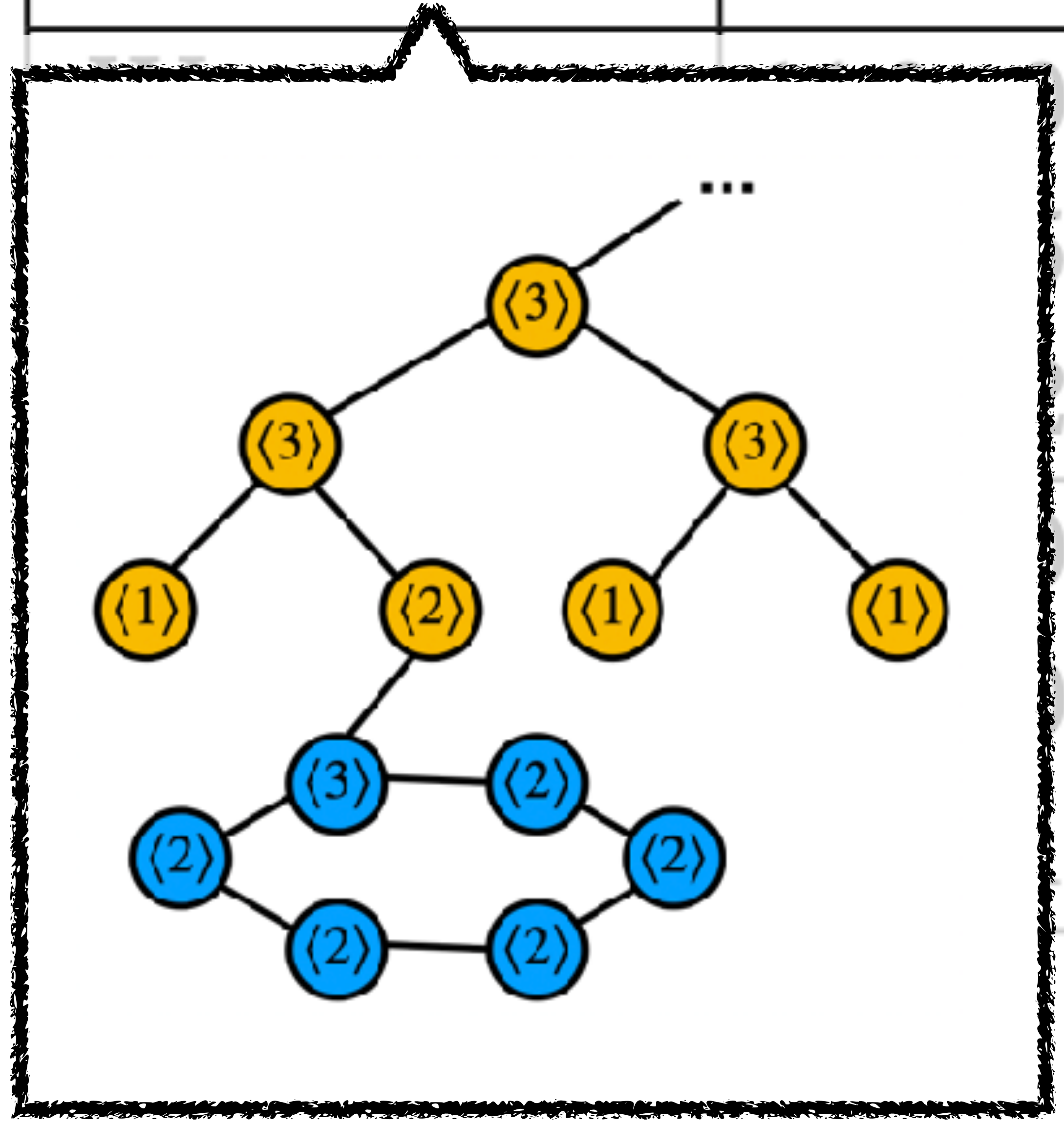
Tree-Cycles 데이터 셋

- 인위적으로 제작한 노드 분류 데이터셋
- 총 2종류(🟡 🔵)의 노드가 있음
  - 각 노드는 degree를 특질로 가짐
  - 🟡 는 8-level 이진 트리의 노드
  - 🔵 는6개 노드로 구성된 사이클의 노드





	GCN							
MUTAG	80.0±0.0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BBBP	83.6±1.4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BACE	78.4±2.8	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
HIV	96.4±0.0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
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**
[illegible]	[illegible]	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
[illegible]	[illegible]	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
[illegible]	[illegible]	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**
[illegible]	[illegible]	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
[illegible]	[illegible]	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
[illegible]	[illegible]	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



(RQ 1) 정확도 비교

	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	[illegible]
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

1~3 등

꼴등

(RQ 1) 정확도 비교

	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	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Wisconsin	64.0±0.0	49.6	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Texas	67.7±5.3	50.0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Cornell	58.9±2.6	61.1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Cora	85.6±0.3	86.4	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
Citeseer	75.2±0.0	74.3	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
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

가장 큰 데이터 셋인 HIV에서는 학습에 실패함

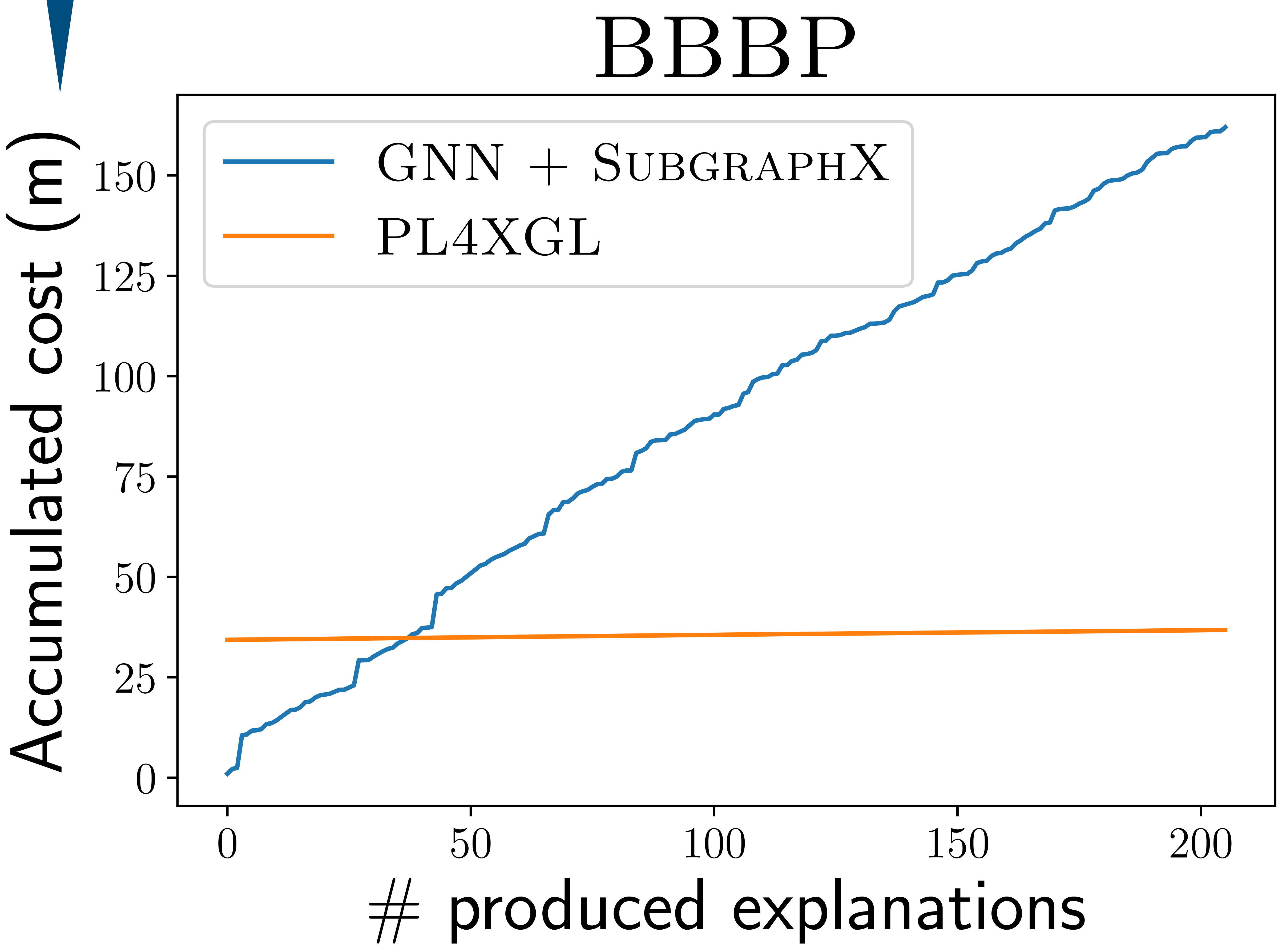
- HIV는 노드 1049163개로 이루어진 데이터셋
- 실패 기준은 48시간 내에 학습이 끝나는지 여부

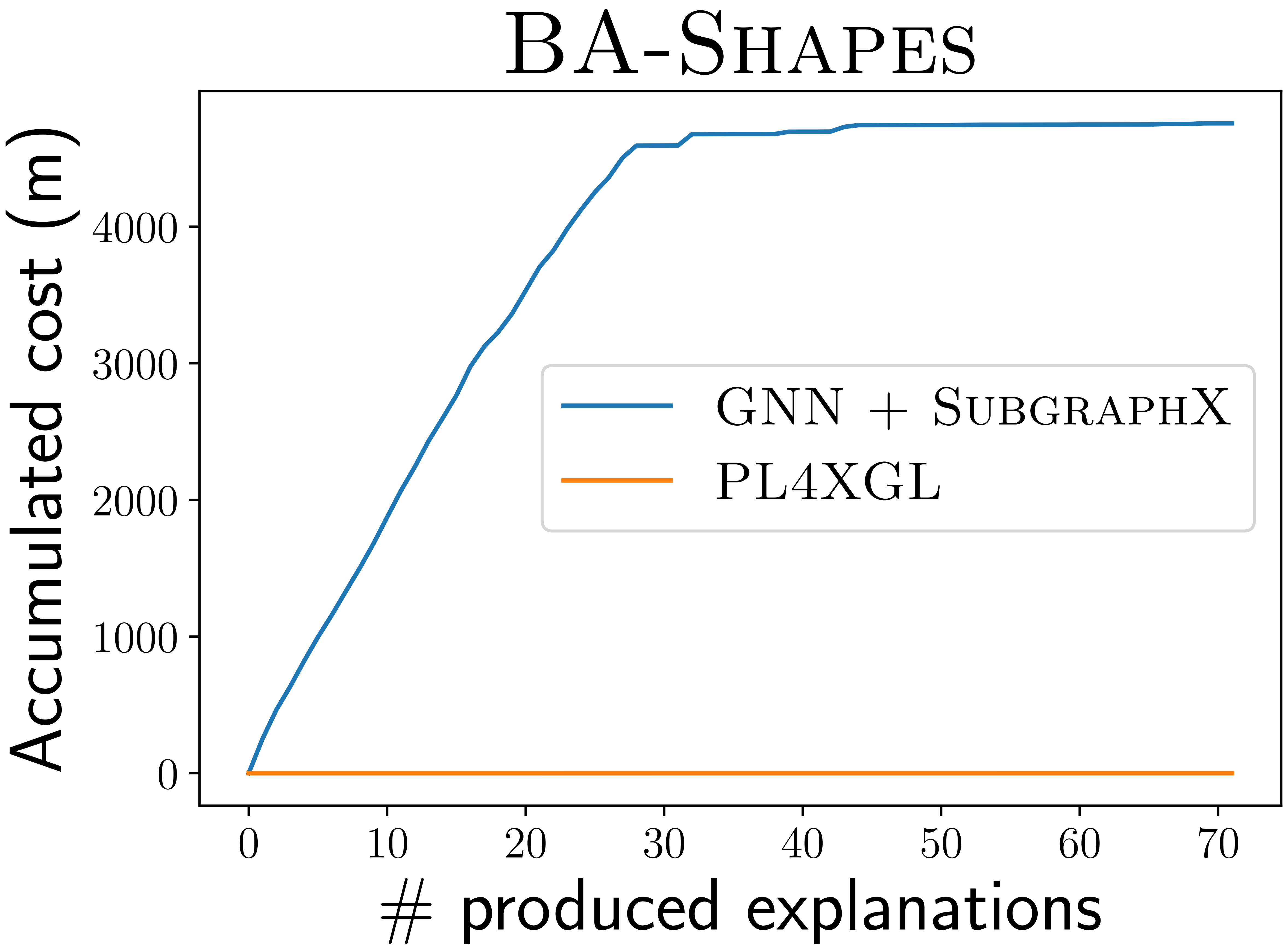
- (RQ 1) GNN과 비교해 정확도는 비등비등함

	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

(RQ 2) 누적(학습+분류+설명) 비용 비교

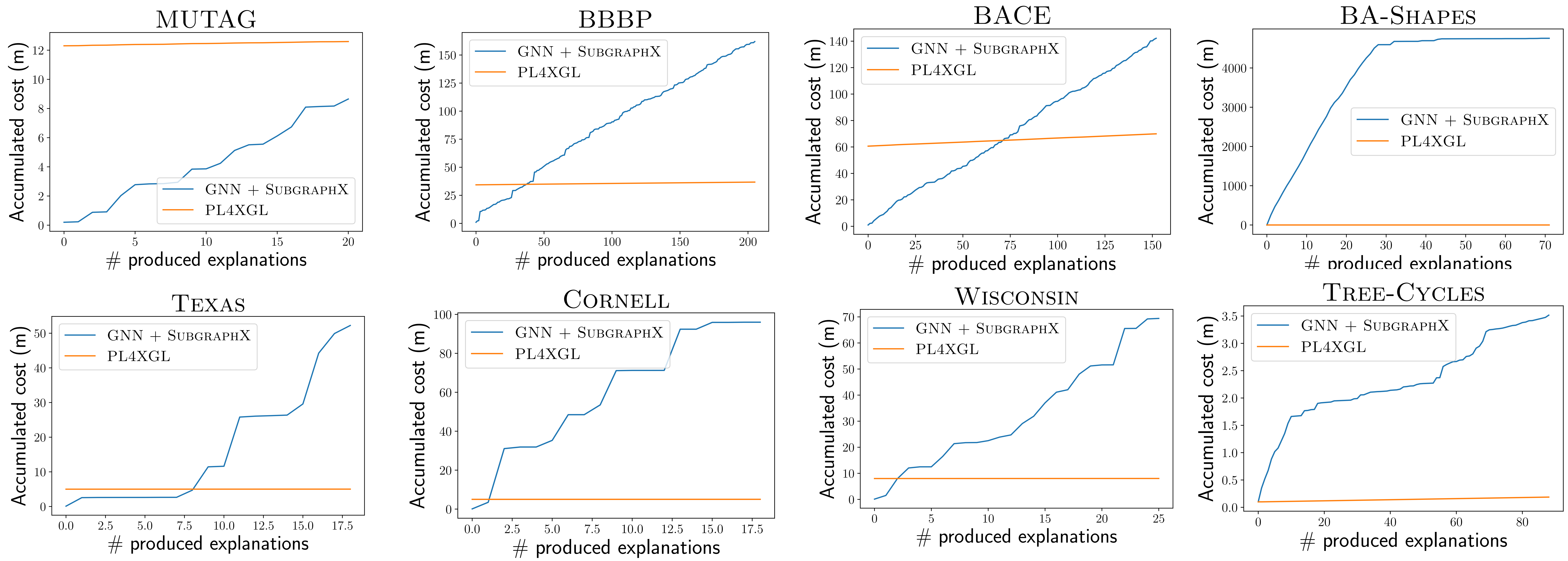
학습 비용 + 분류 비용 + 설명 비용





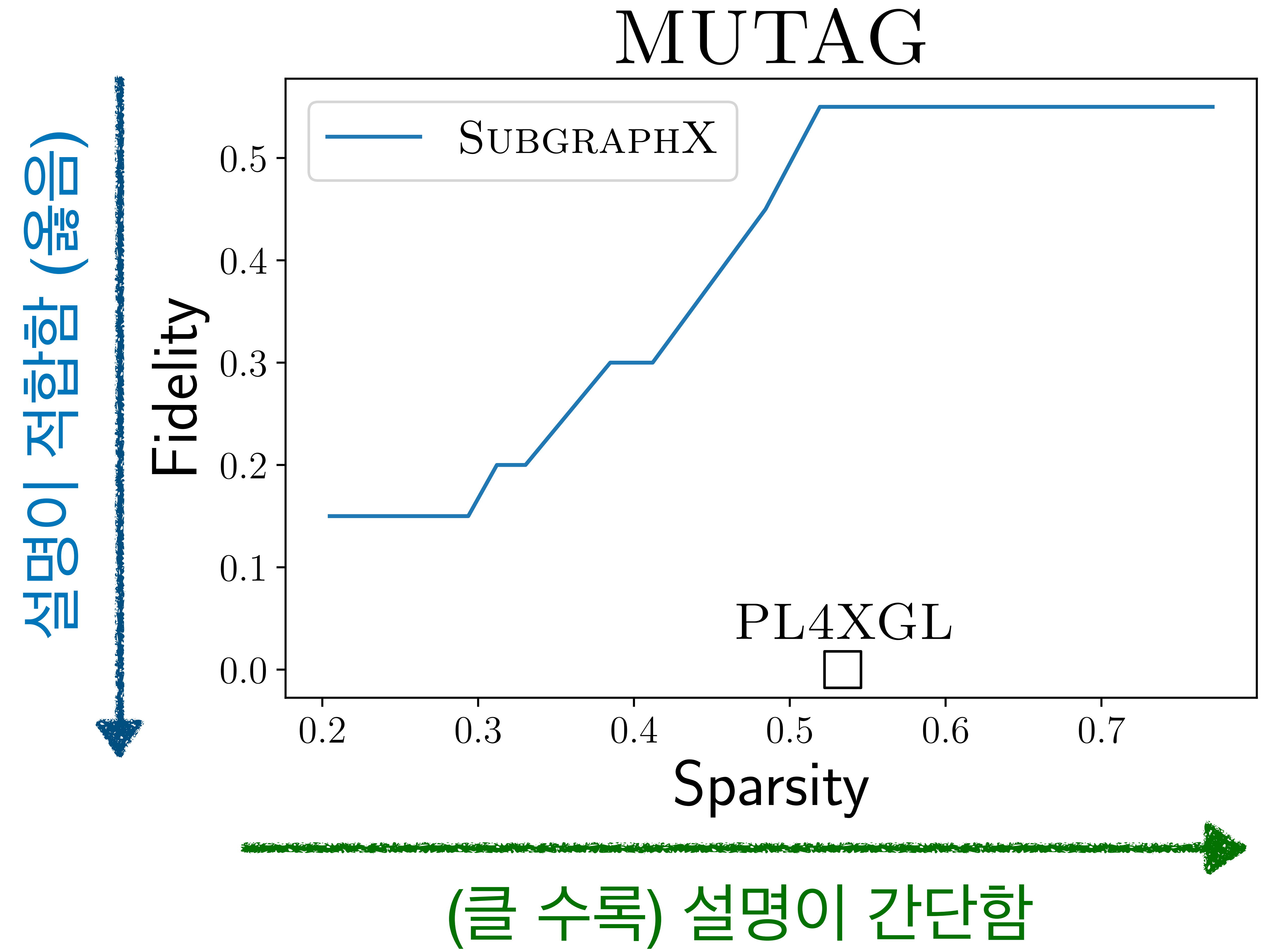
- (RQ 1) GNN+SubgraphX와 비교해 빠름

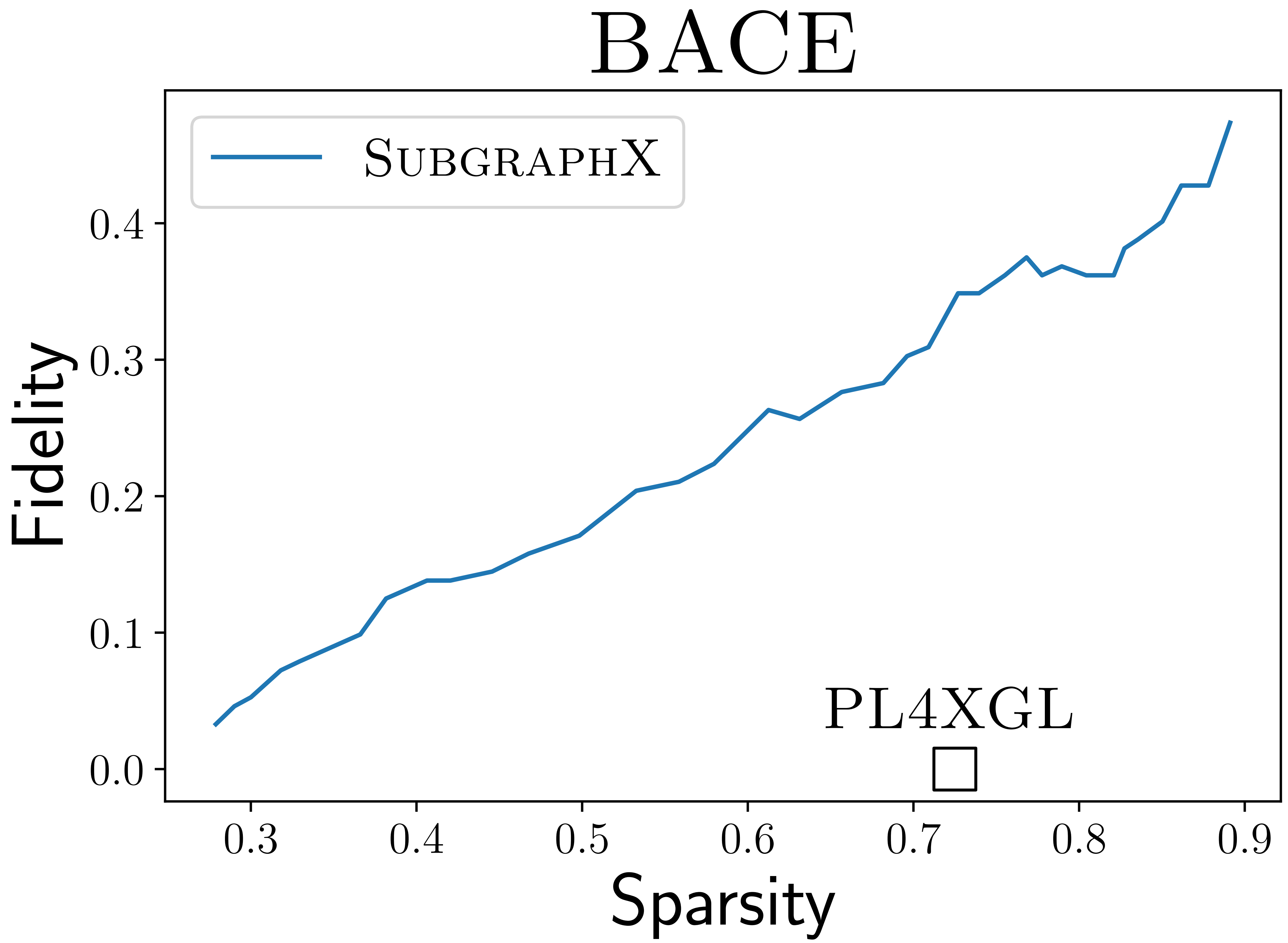
: PL4XGL
: GNN + SubgraphX



(RQ 3) 제공된 설명의 품질 비교

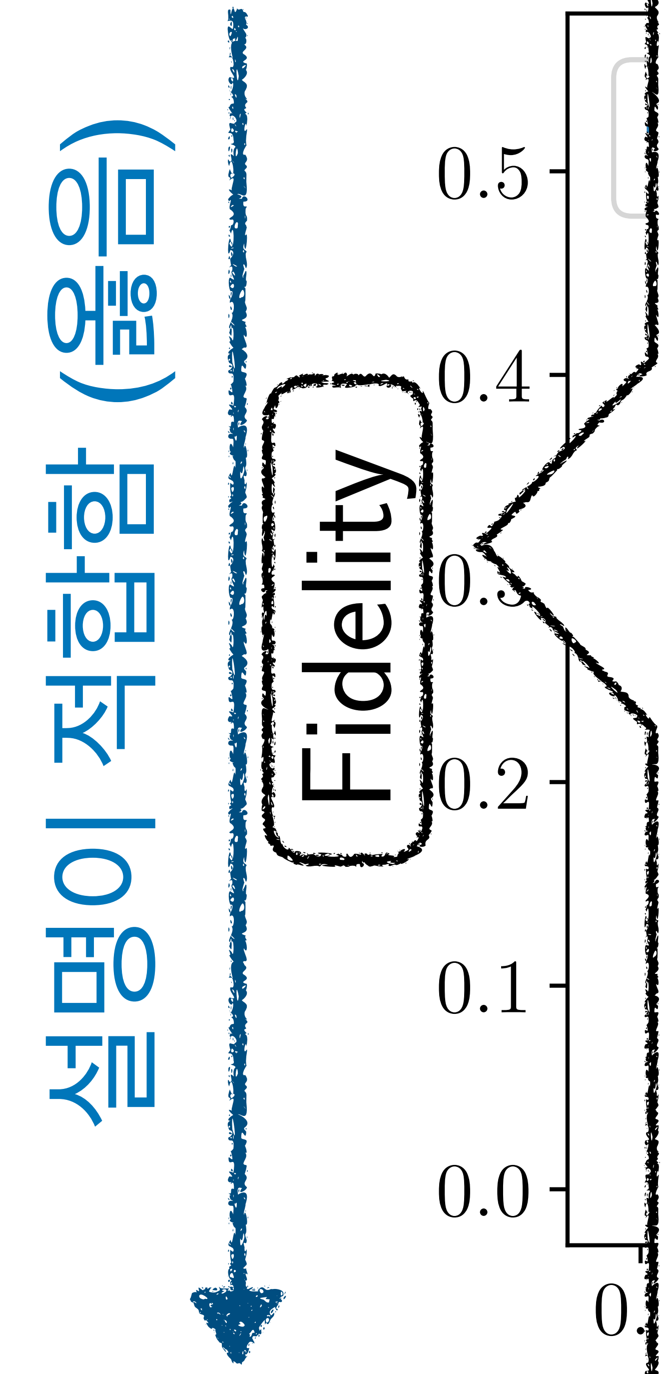
- 설명의 적합성(fidelity), 간단성(Sparsity) 비교



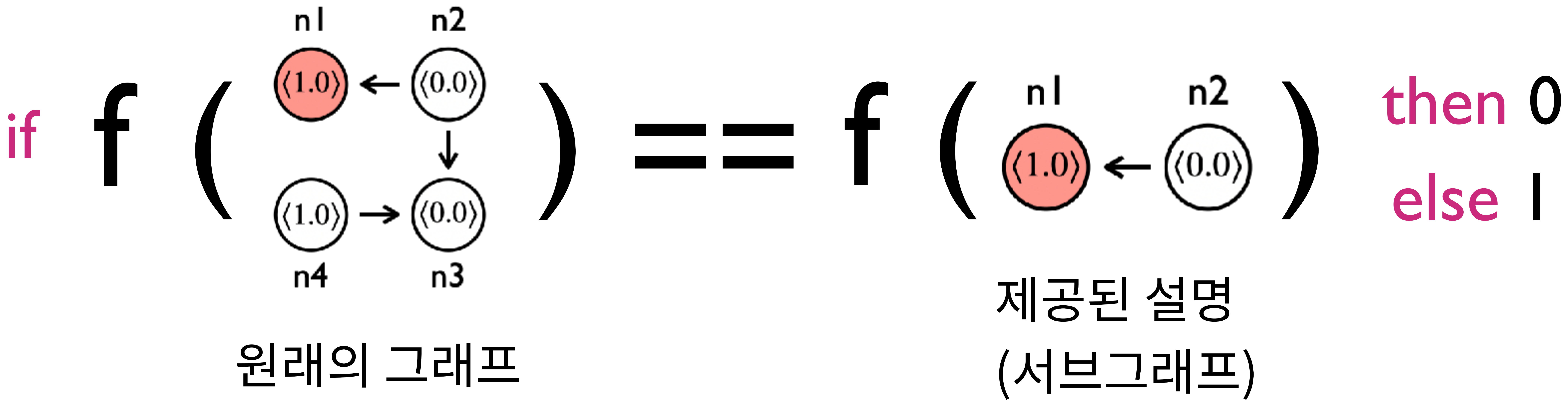


(R.Q 3) 제공된 설명의 품질 비교

- 설명의 적



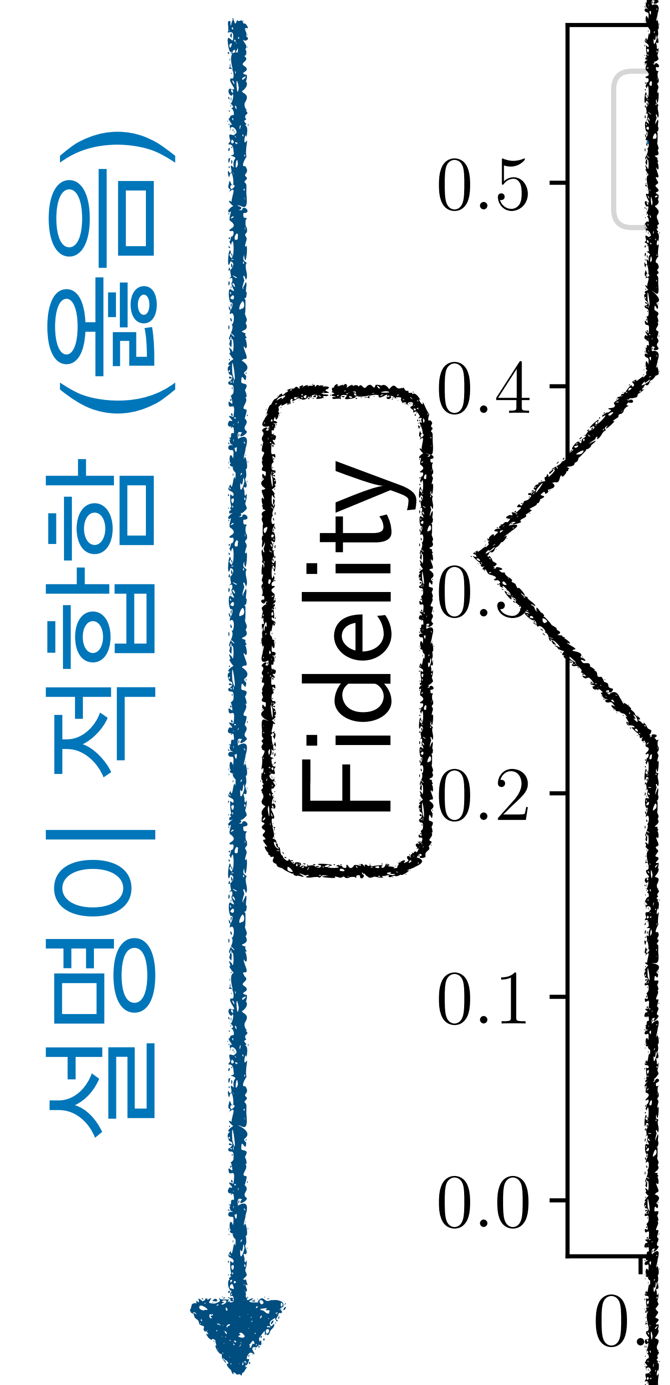
- Fidelity(충실성)의 목표는 서브그래프 설명이 얼마나 모델에 충실한지를 비교
  - 모델(f)이 제동된 서브그래프를 원래 그래프랑 같은 레이블로 분류하는지 측정



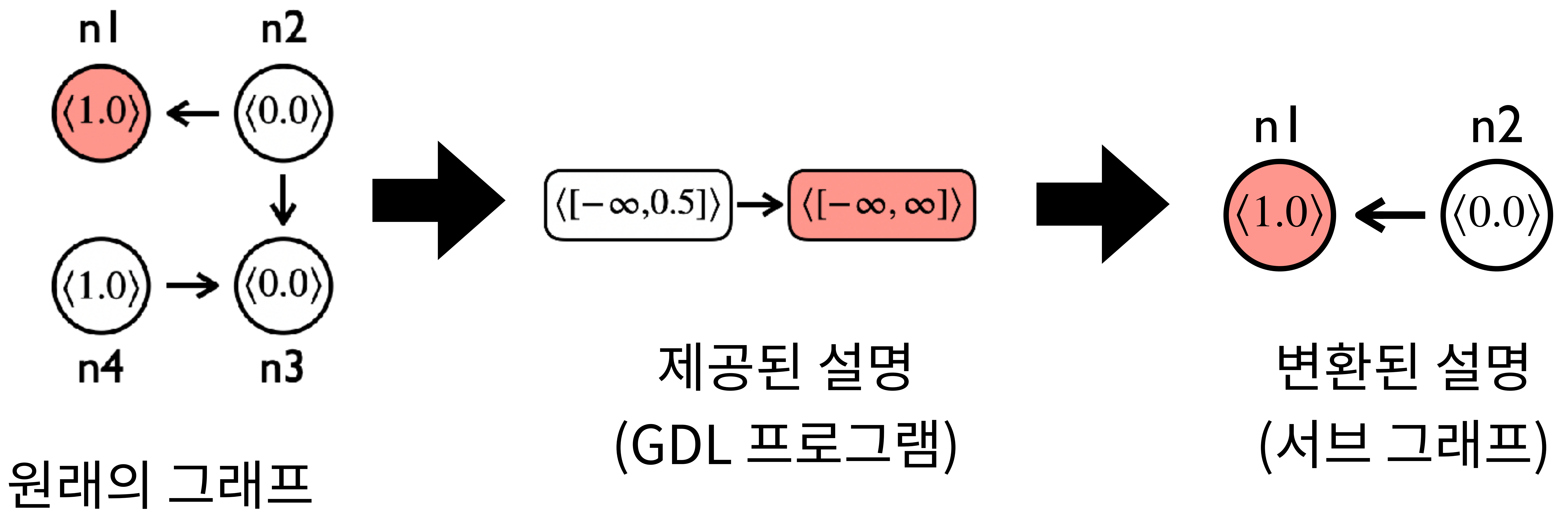
(클 수록) 설명이 간단함

(RQ 3) 제공된 설명의 품질 비교

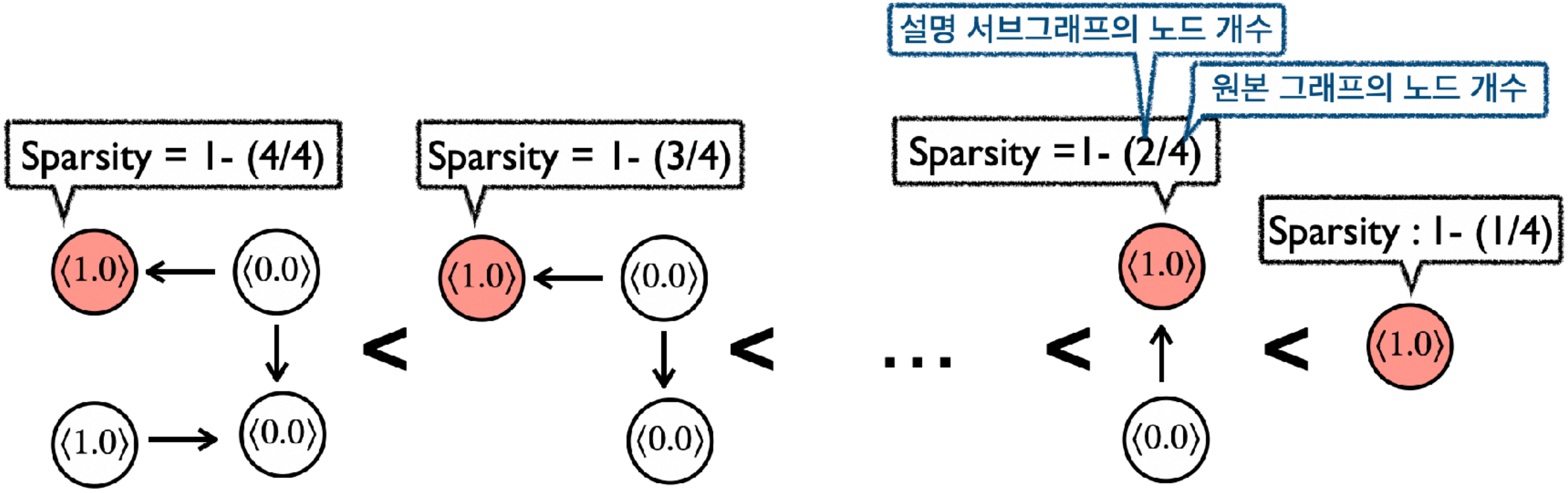
• 설명의 적



• GDL 프로그램은 서브그래프로 변환하여 Fidelity를 측정함



- Sparsity: 설명이 간단할 수록 좋은 설명이다

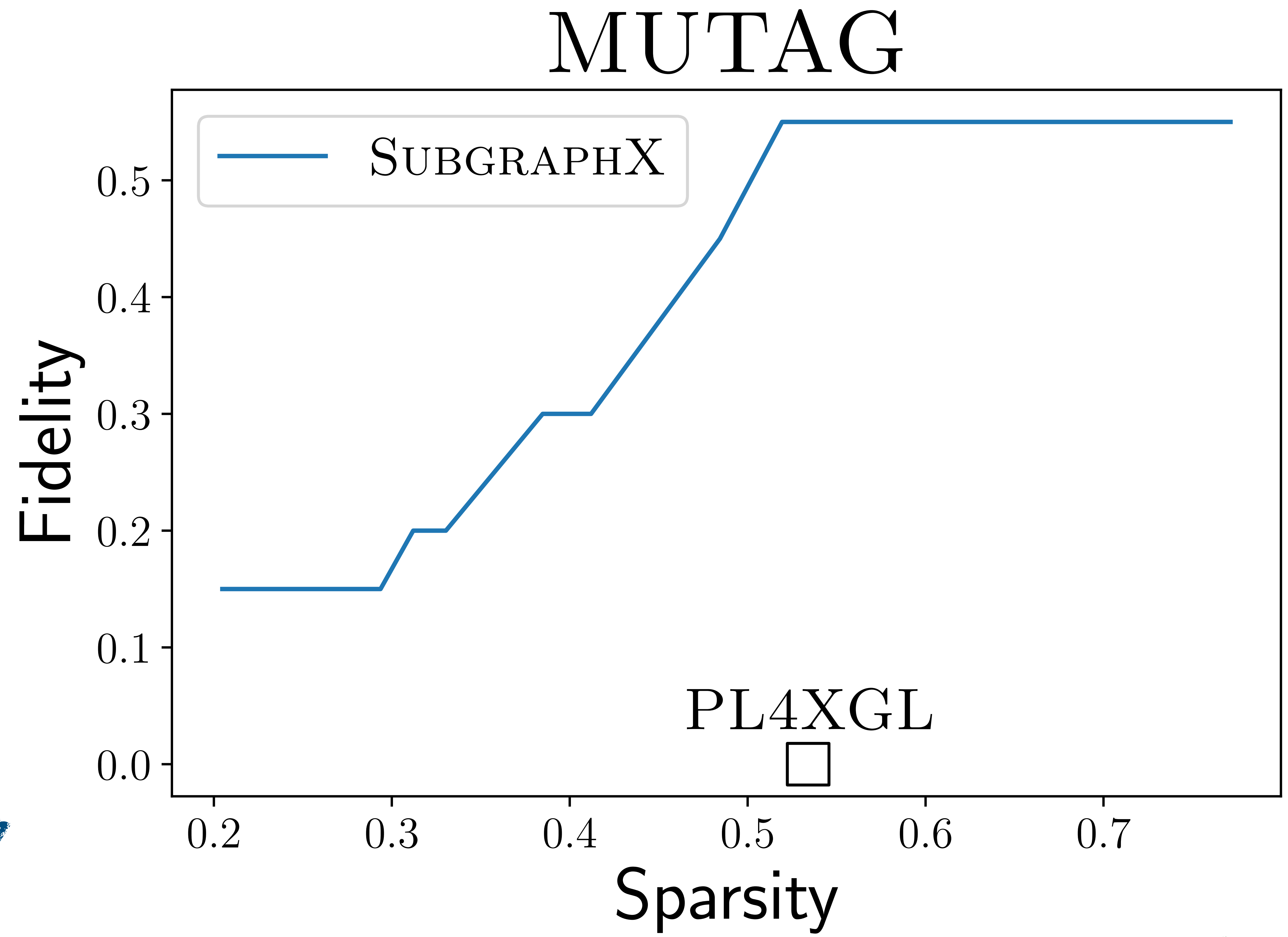


(클 수록) 설명이 간단함

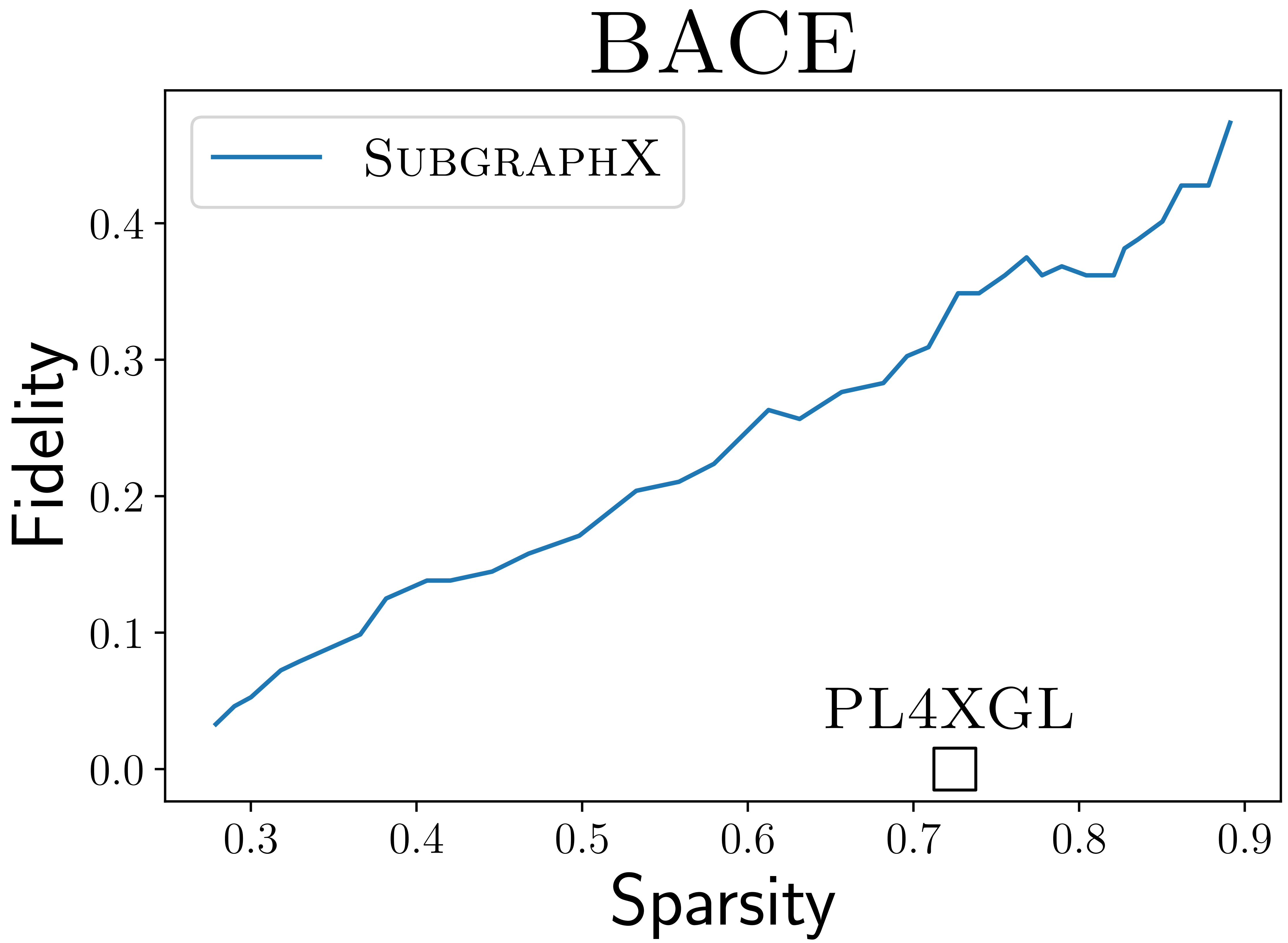
(RQ 3) 제공된 설명의 품질 비교

- 설명의 적합성(fidelity), 간단성(Sparsity) 비교

설명이 적합함 (옳음)



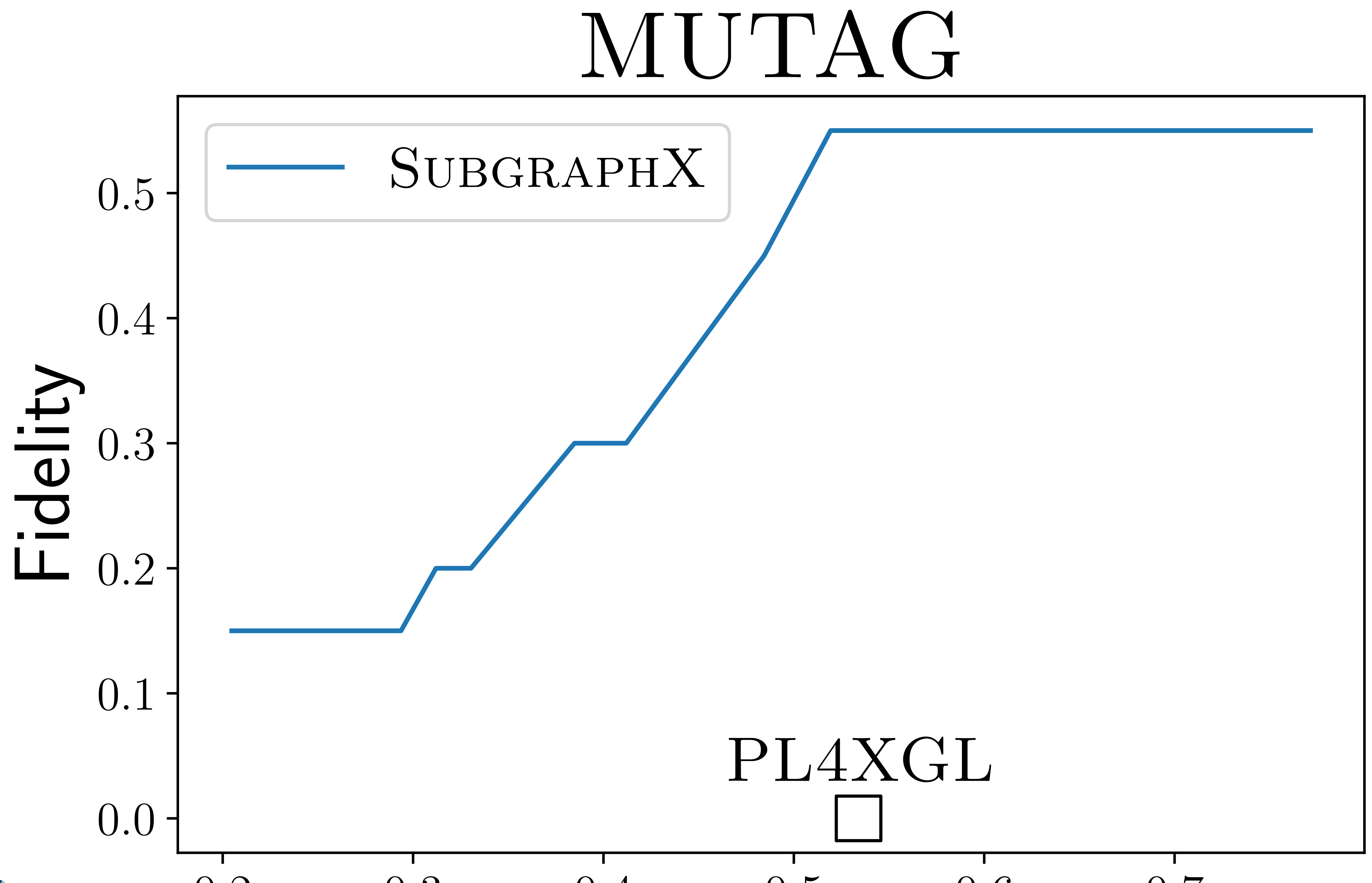
(클 수록) 설명이 간단함

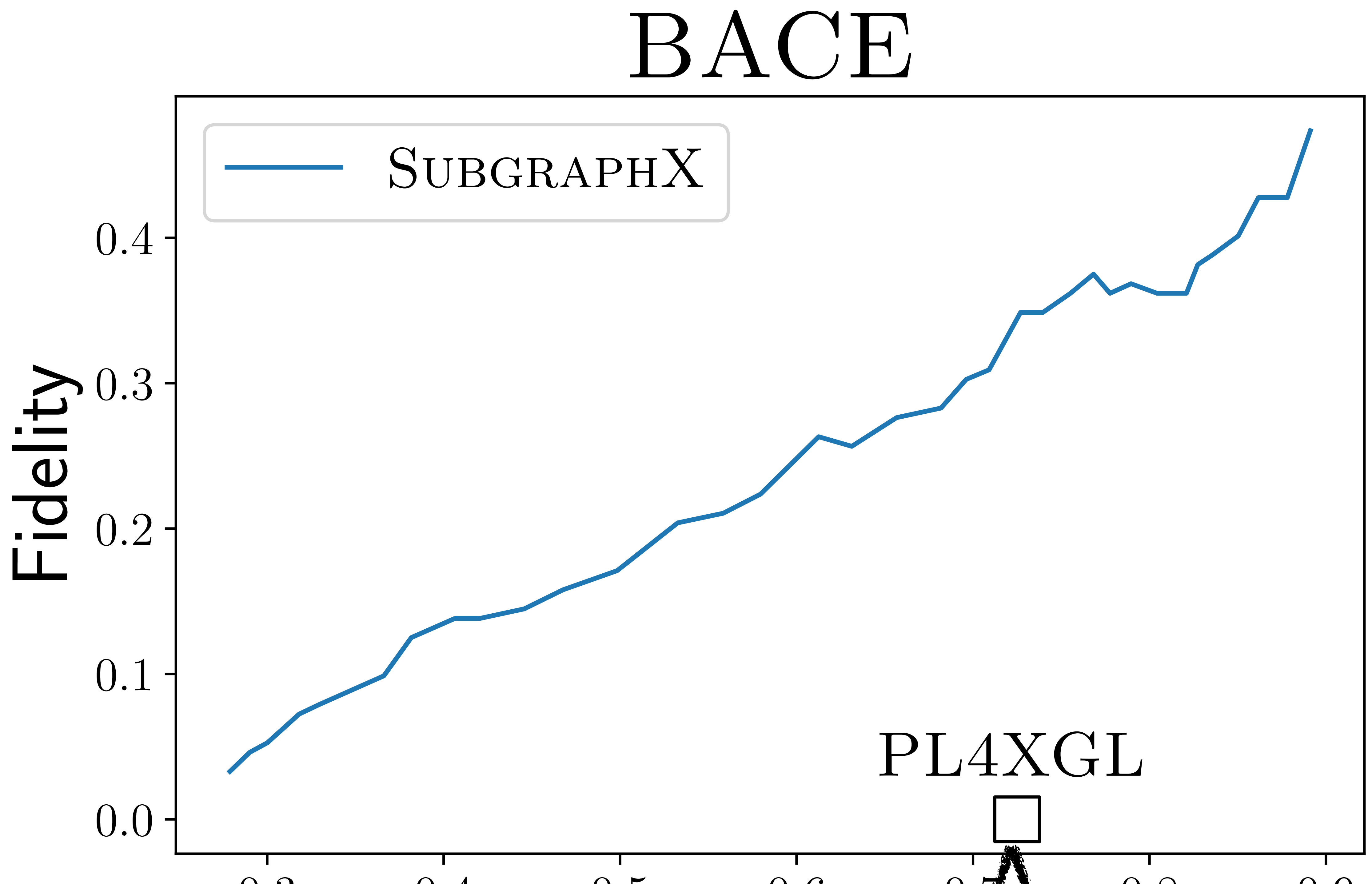


(RQ 3) 제공된 설명의 품질 비교

- 설명의 적합성(fidelity), 간단성(Sparsity) 비교

설명이 적합함 (옳음)

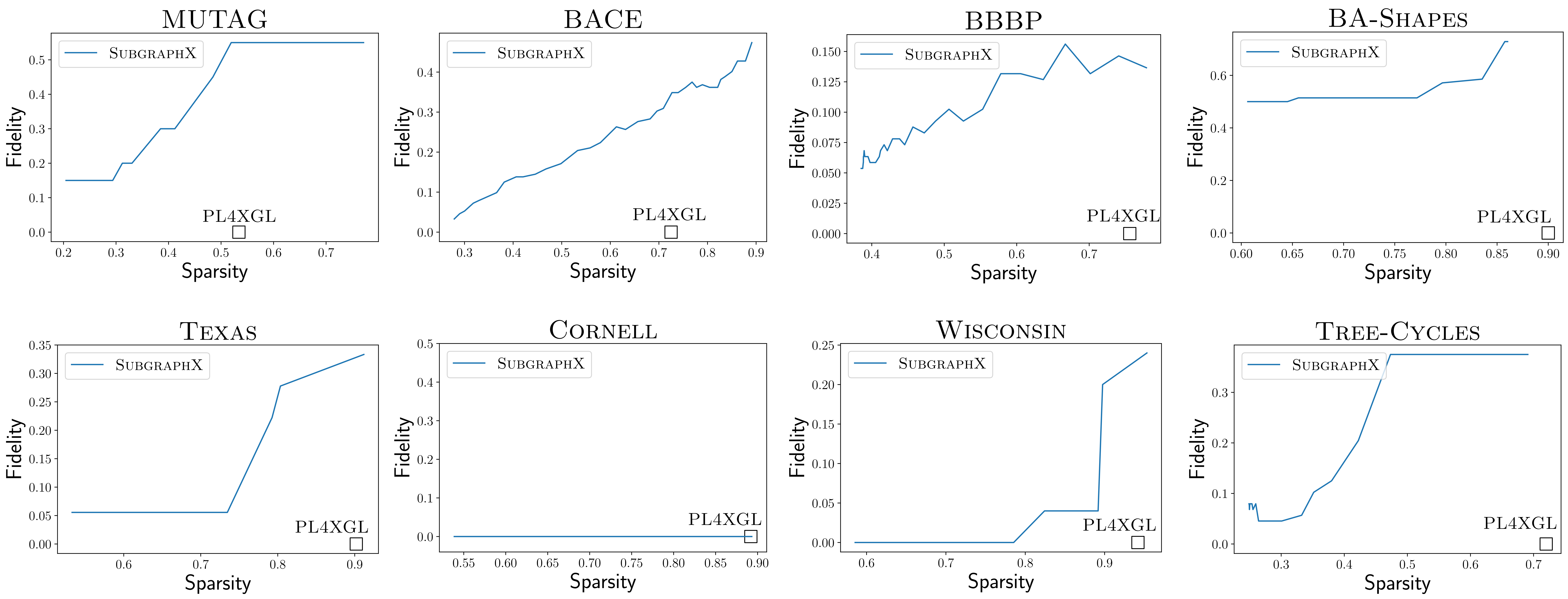




항상 Fidelity = 0.0 임이 보장됨

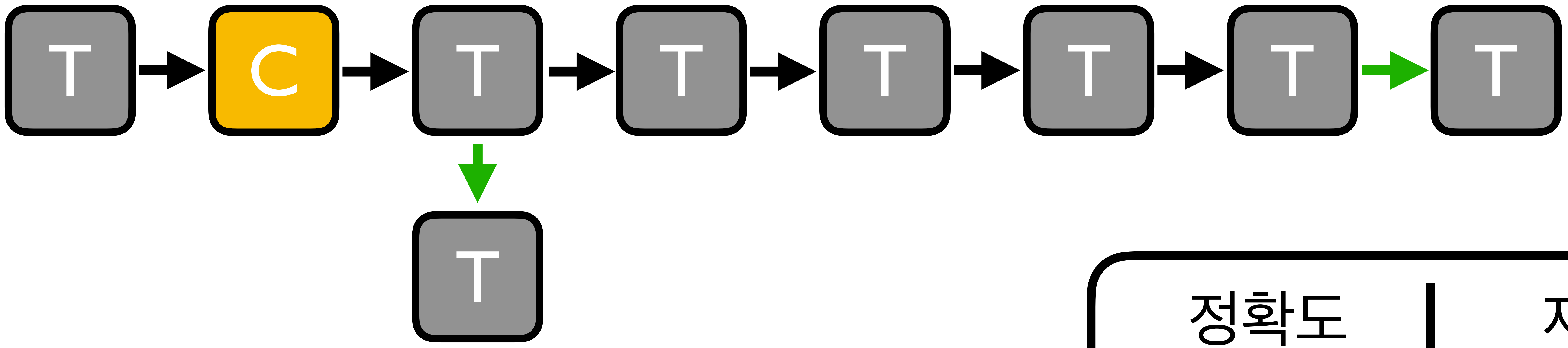
THEOREM 6.1. *If PL4XGL classifies a graph G into a label i and provides a GDL program P as an explanation, PL4XGL classifies all the subgraphs transformed from P into the same label i.*

(RQ 3) 제공된 설명의 품질 비교



학습된 고품질 GDL 프로그램

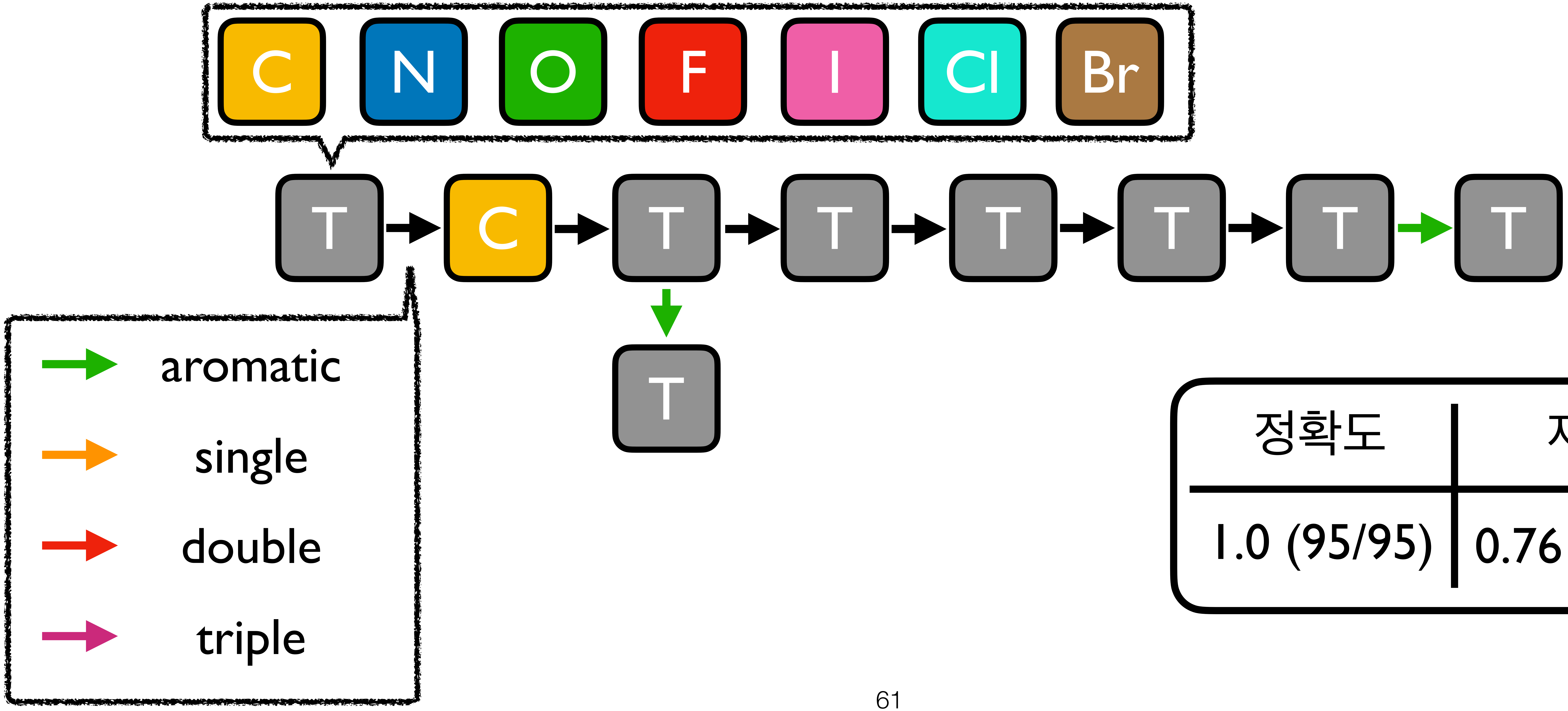
- Salmonella 박테이라 반응이 양성인 분자의 패턴 (MUTAG 데이터 셋)



정확도	재현율
1.0 (95/95)	0.76 (95/125)

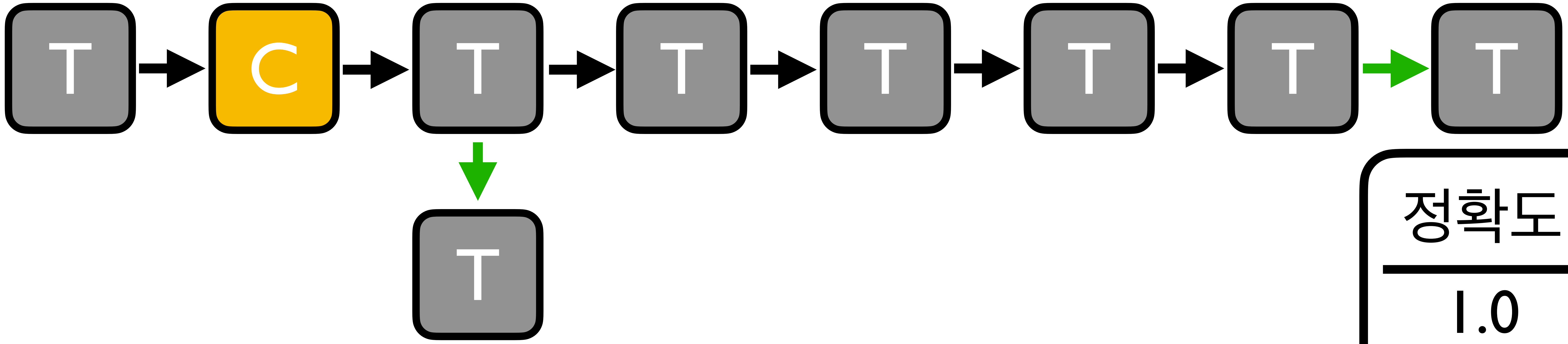
학습된 고품질 GDL 프로그램

- Salmonella 박테이라 반응이 양성인 분자의 패턴 (MUTAG 데이터 셋)

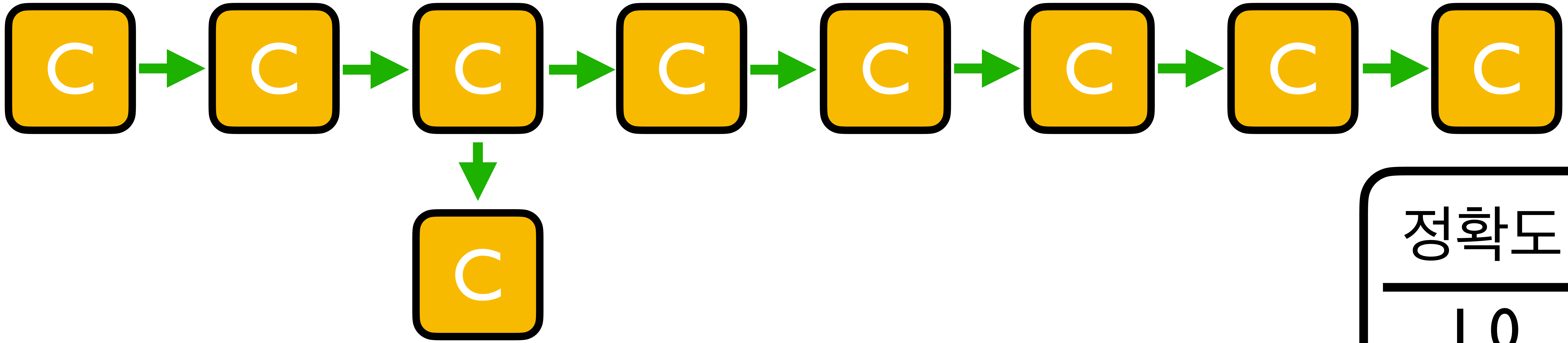


정확도	재현율
1.0 (95/95)	0.76 (95/125)

GDL 프로그램 vs 서브그래프



정확도	재현율
1.0	0.76



정확도	재현율
1.0	0.36

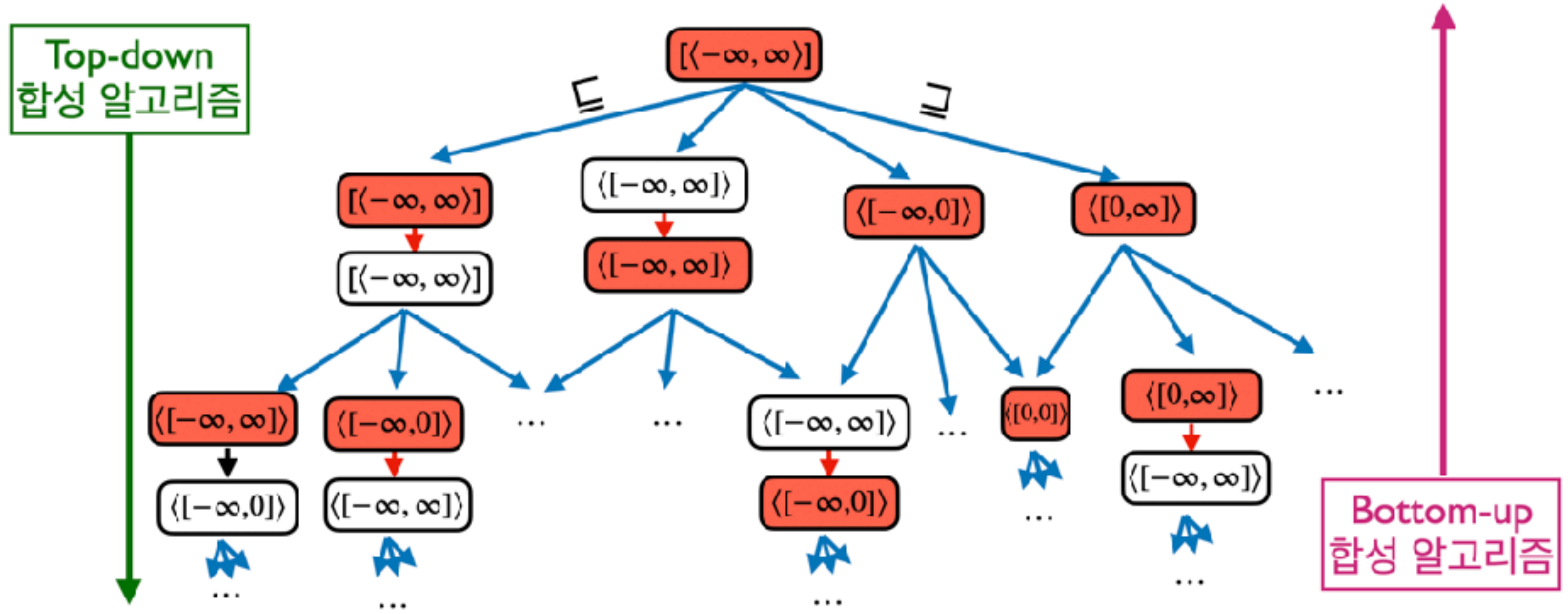
결론

- 프로그래밍 언어 기술을 이용하여 본질적으로 설명 가능한 그래프 기계학습 방법을 개발함
  - 핵심 아이디어 1: 그래프 패턴 표현 프로그래밍 언어 (GDL)
  - 핵심 아이디어 2: 그래프 패턴 프로그램 자동 합성 알고리즘

그래프 패턴 표현 프로그래밍 언어 (GDL)

Programs	P	$::= \overline{\delta}\ \texttt{target}\ t$
Descriptions	δ	$::= \delta_V \mid \delta_E$
Node Descriptions	δ_V	$::= \texttt{node}\ x\ \texttt{<}\overline{\phi}\texttt{>}^?$
Edge Descriptions	δ_E	$::= \texttt{edge}\ (x,x)\ \texttt{<}\overline{\phi}\texttt{>}^?$
Target Symbols	t	$::= \texttt{node}\ x \mid \texttt{edge}\ (x,x) \mid \texttt{graph}$
Intervals	ϕ	$::= [n^?, n^?]$
Real Numbers	n	$::= \texttt{0.2} \mid \texttt{0.7} \mid \texttt{6} \mid \texttt{-8} \ldots$
Variables	x	$::= \texttt{x} \mid \texttt{y} \mid \texttt{z} \mid \ldots$

그래프 패턴 프로그램 자동 합성 알고리즘



OOPSLA' 20

데이터 기반 정적 분석을 위한 특질 자동 생성

In progress

결함 위치 추정 (Fault localization)

In progress

그래프 패턴 언어 및 합성 알고리즘 개선

Graph Description Language

Programs	P	::=	δ **target** t
Descriptions	δ	::=	$\delta_V \mid \delta_E$
Node Descriptions	δ_V	::=	**node** x <$\overline{\phi}$>$^?$
Edge Descriptions	δ_E	::=	**edge** (x, x) <$\overline{\phi}$>$^?$
Target Symbols	t	::=	**node** x | **edge** (x, x) | **graph**
Intervals	ϕ	::=	$[n^?, n^?]$
Real Numbers	n	::=	0.2 | 0.7 | 6 | -8 ...
Variables	x	::=	x | y | z | ...

PLDI' 24

설명 가능한 그래프 기계학습 방법

In progress

GDL 기반 Graph Isomorphism

ToDo

GDL 기반 그래프 데이터 마이닝

ToDo

GDL 기반 GNN 설명 기법

감사합니다!