될 때까지 개선하기

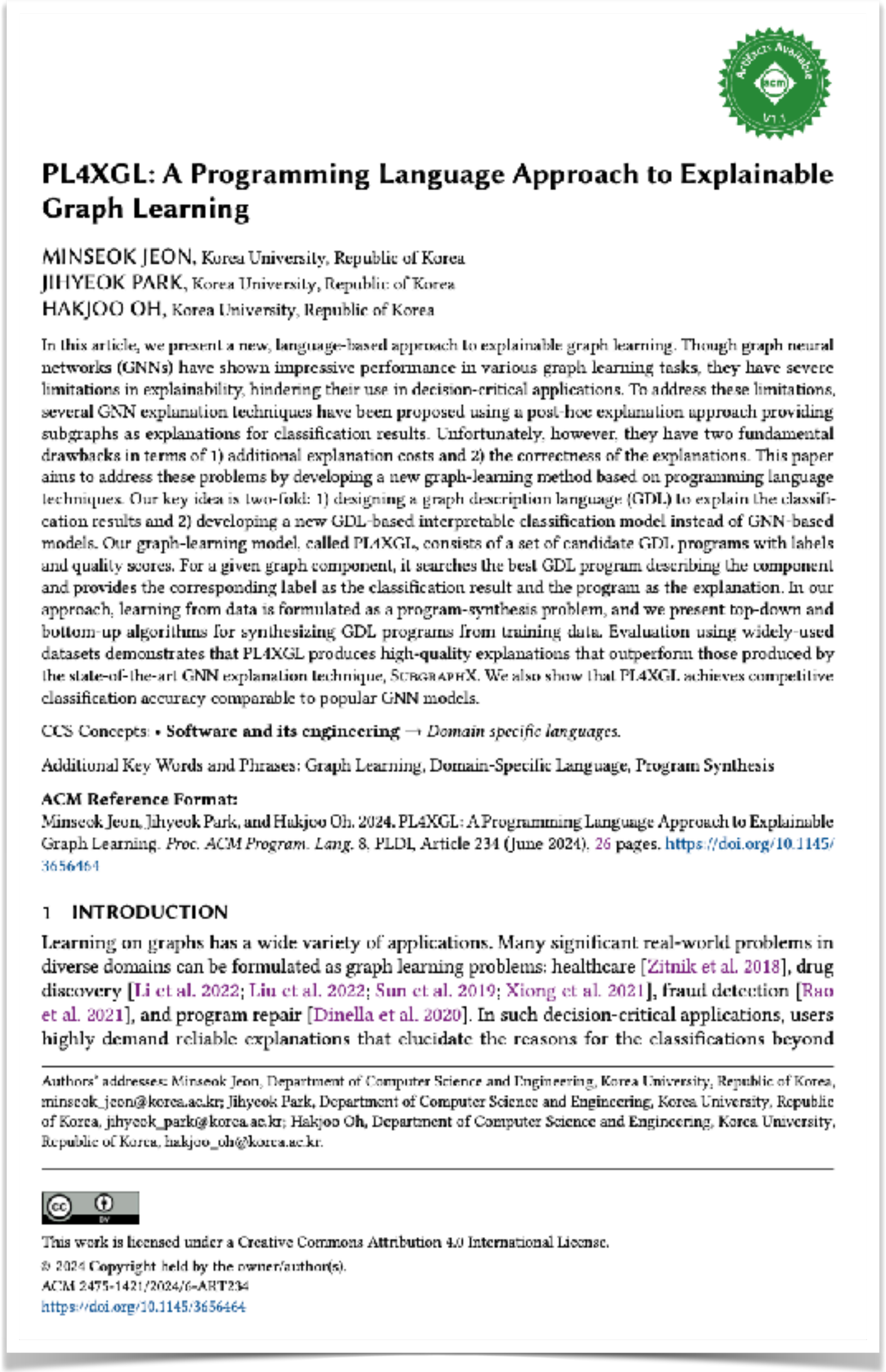
(PL기술로 새로운 기계학습 방법 개발하기)

전민석

KOREA UNIVERSITY

Aug.23.2024 @ SIGPL 여름학교

될 때까지 개선하기



PL4XGL: A Programming Language Approach to Explainable Graph Learning

MINSEOK JEON, Korea University, Republic of Korea
JIHYEOK PARK, Korea University, Republic of Korea
HAKJOO OH, Korea University, Republic of Korea

In this article, we present a new, language-based approach to explainable graph learning. Though graph neural networks (GNNs) have shown impressive performance in various graph learning tasks, they have severe limitations in explainability, hindering their use in decision-critical applications. To address these limitations, several GNN explanation techniques have been proposed using a post-hoc explanation approach providing subgraphs as explanations for classification results. Unfortunately, however, they have two fundamental drawbacks in terms of 1) additional explanation costs and 2) the correctness of the explanations. This paper aims to address these problems by developing a new graph-learning method based on programming language techniques. Our key idea is two-fold: 1) designing a graph description language (GDL) to explain the classification results and 2) developing a new GDL-based interpretable classification model instead of GNN-based models. Our graph-learning model, called PL4XGL, consists of a set of candidate GDL programs with labels and quality scores. For a given graph component, it searches the best GDL program describing the component and provides the corresponding label as the classification result and the program as the explanation. In our approach, learning from data is formulated as a program-synthesis problem, and we present top-down and bottom-up algorithms for synthesizing GDL programs from training data. Evaluation using widely-used datasets demonstrates that PL4XGL produces high-quality explanations that outperform those produced by the state-of-the-art GNN explanation technique, SUBGRAPHX. We also show that PL4XGL achieves competitive classification accuracy comparable to popular GNN models.

CCS Concepts: • **Software and its engineering** → *Domain specific languages*.

Additional Key Words and Phrases: Graph Learning, Domain-Specific Language, Program Synthesis

ACM Reference Format:
Minseok Jeon, Jihyeok Park, and Hakjoo Oh. 2024. PL4XGL: A Programming Language Approach to Explainable Graph Learning. *Proc. ACM Program. Lang.* 8, PLDI, Article 234 (June 2024), 26 pages. https://doi.org/10.1145/3656464

1 INTRODUCTION

Learning on graphs has a wide variety of applications. Many significant real-world problems in diverse domains can be formulated as graph learning problems: healthcare [Zitnik et al. 2018], drug discovery [Li et al. 2022; Liu et al. 2022; Sun et al. 2019; Xiong et al. 2021], fraud detection [Rao et al. 2021], and program repair [Dinella et al. 2020]. In such decision-critical applications, users highly demand reliable explanations that elucidate the reasons for the classifications beyond

Authors' addresses: Minseok Jeon, Department of Computer Science and Engineering, Korea University, Republic of Korea, minseok_jeon@korea.ac.kr; Jihyeok Park, Department of Computer Science and Engineering, Korea University, Republic of Korea, jihyeok_park@korea.ac.kr; Hakjoo Oh, Department of Computer Science and Engineering, Korea University, Republic of Korea, hakjoo_oh@korea.ac.kr.

This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2024 Copyright held by the owner/author(s).
ACM 2475-1421/2024/6-ART234
https://doi.org/10.1145/3656464

연구기간: 3년 (2021.01~2023.11)

- ICML2022: Rejected
- NIPS2022: Rejected
- PLDI2023: Rejected
- POPL2024: Rejected
- PLDI2024: Accepted

개선
개선
개선
개선

A New Explainable Machine Learning for Node Classification

Minseok Jeon
Software Analysis Laboratory
Korea University

29 January 2021

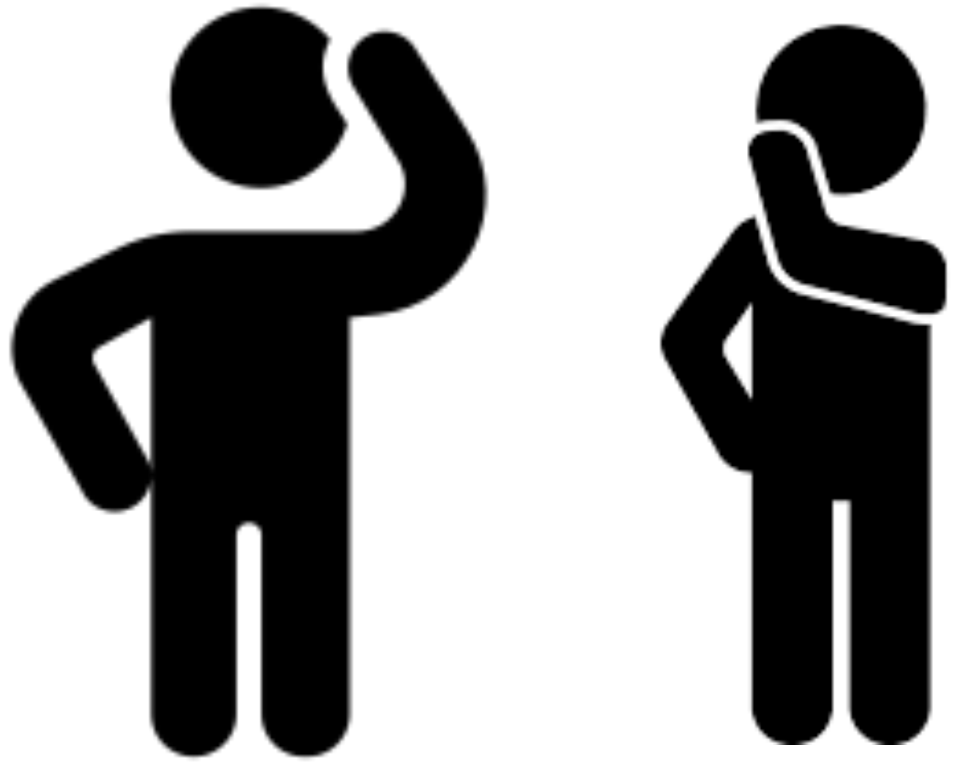
연구 시작 (2021.01)
ICML 제출 (2022.02)
NIPS 제출 (2022.05)
PLDI 제출 (2022.11)
POPL 제출 (2023.07)
PLDI 제출 (2023.11)

개인적인 연구동기

- 머신러닝랩(MLV)과 합동 연구 미팅 중

낮을수록 좋은 분석

	antlr				
	recall	precision	alarms	costs	select
Minimal	-	-	463	28.12	0
Graphick	1.0	0.19	463	26.83	2629
MLP	1.0	0.108	463	91	4731
GCN	1.0	0.07	463	107.74	6183
GCN(Full)	1.0	0.069	X	X	7394
GAT	1.0	0.09	-	-	5645
GAT(Full)	1.0	0.092	-	-	5567
APPNP	1.0	0.069	-	-	7369
GCN(Concat)	1.0	0.109	-	-	4681
GCN(OnlyEdge)	1.0	0.109	-	-	4710
GCN(Primitive)	1.0	0.160	463	57.12	3194



개인적인 연구동기

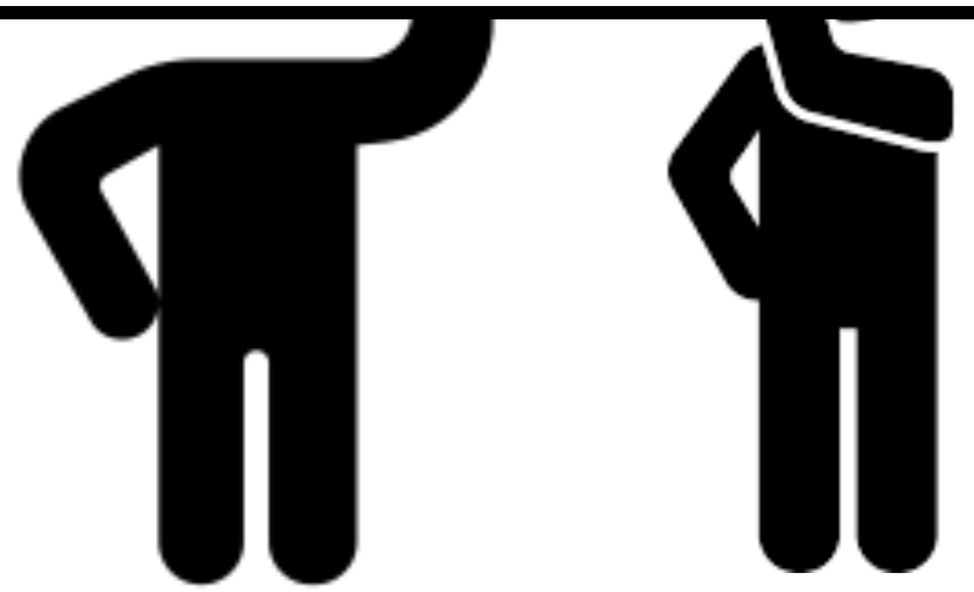
- 머신러닝랩(MI

Learning Graph-Based Heuristics for Pointer Analysis without Handcrafting Application-Specific Features

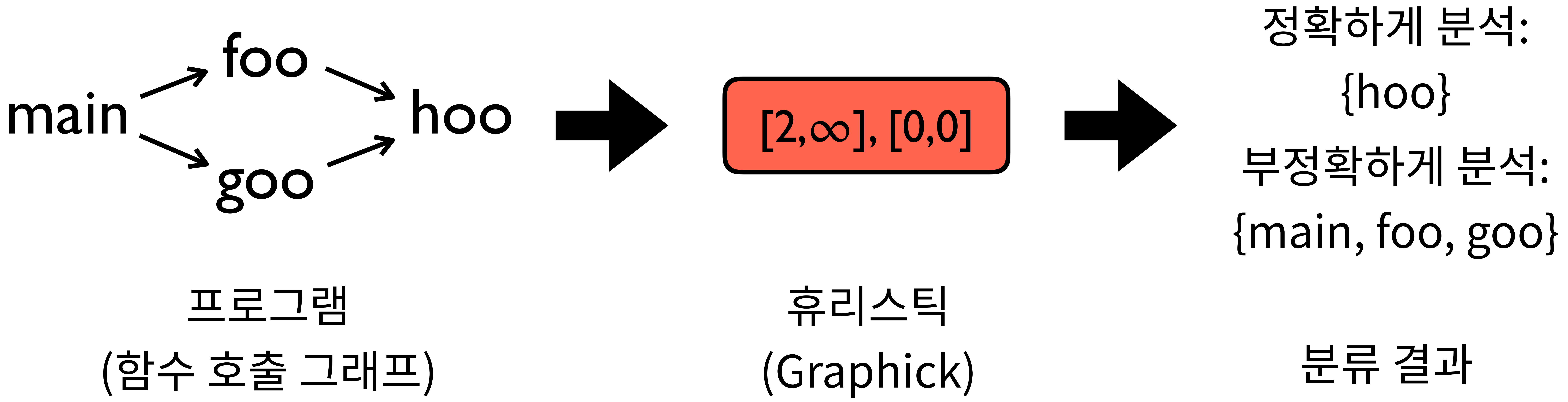
MINSEOK JEON, MYUNGHO LEE, and HAKJOO OH*, Korea University, Republic of Korea

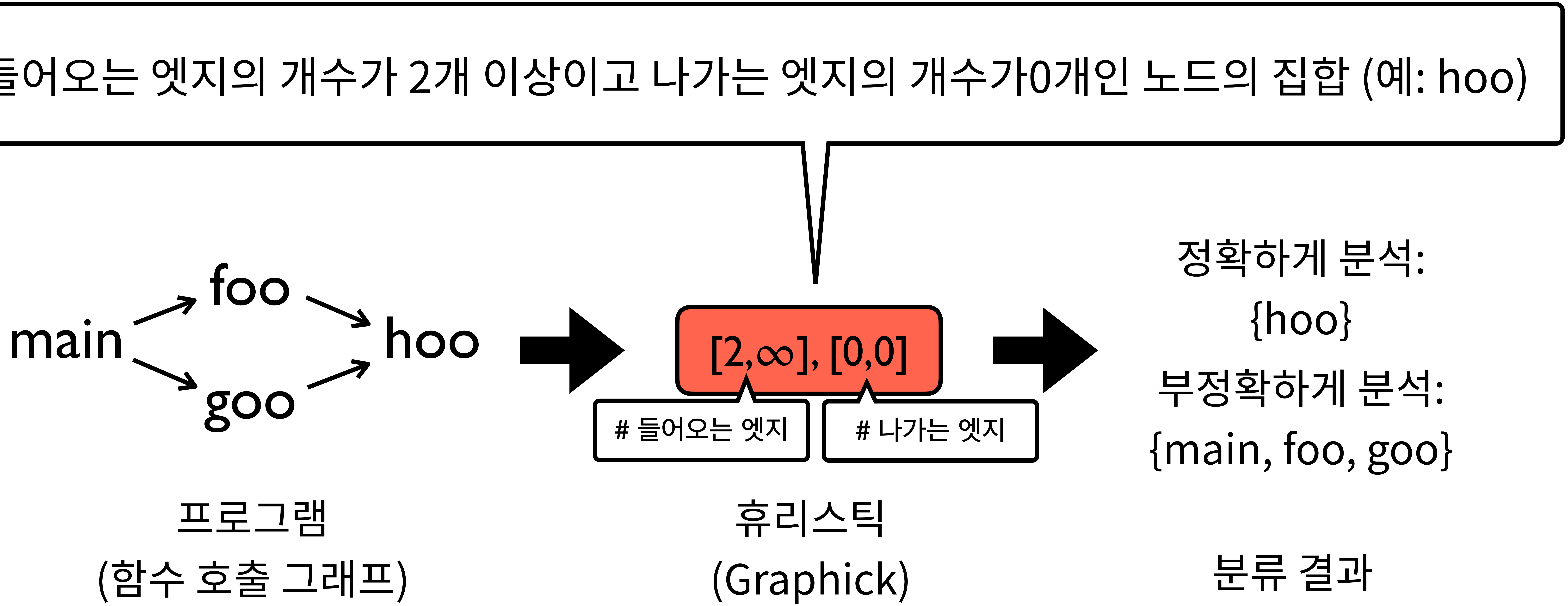
OOPSLA 2020

Minimal					
Graphick					
MLP					
GCN	1.0	[illegible]	[illegible]	[illegible]	[illegible]
GCN(Full)	1.0	0.069	X	X	7394
GAT	1.0	0.09	-	-	5645
GAT(Full)	1.0	0.092	-	-	5567
APPNP	1.0	0.069	-	-	7369
GCN(Concat)	1.0	0.109	-	-	4681
GCN(OnlyEdge)	1.0	0.109	-	-	4710
GCN(Primitive)	1.0	0.160	463	57.12	3194

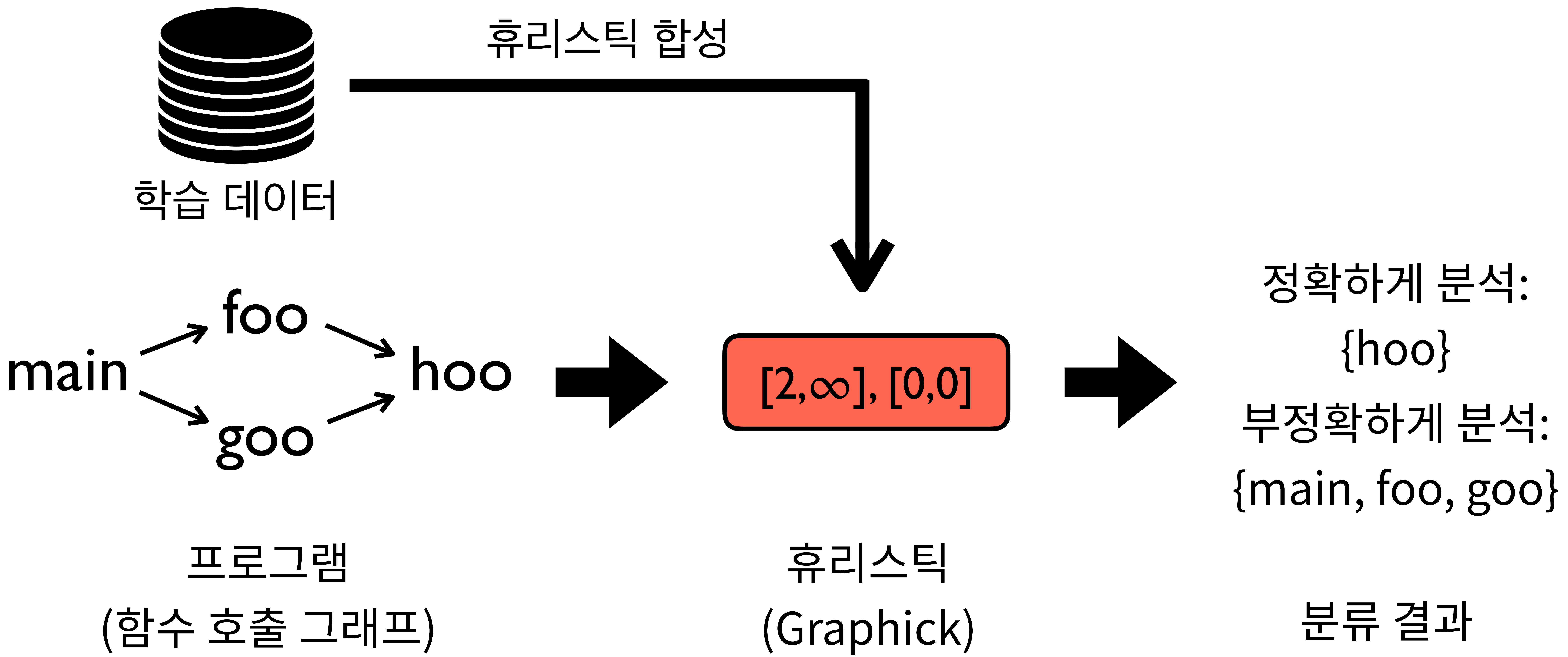


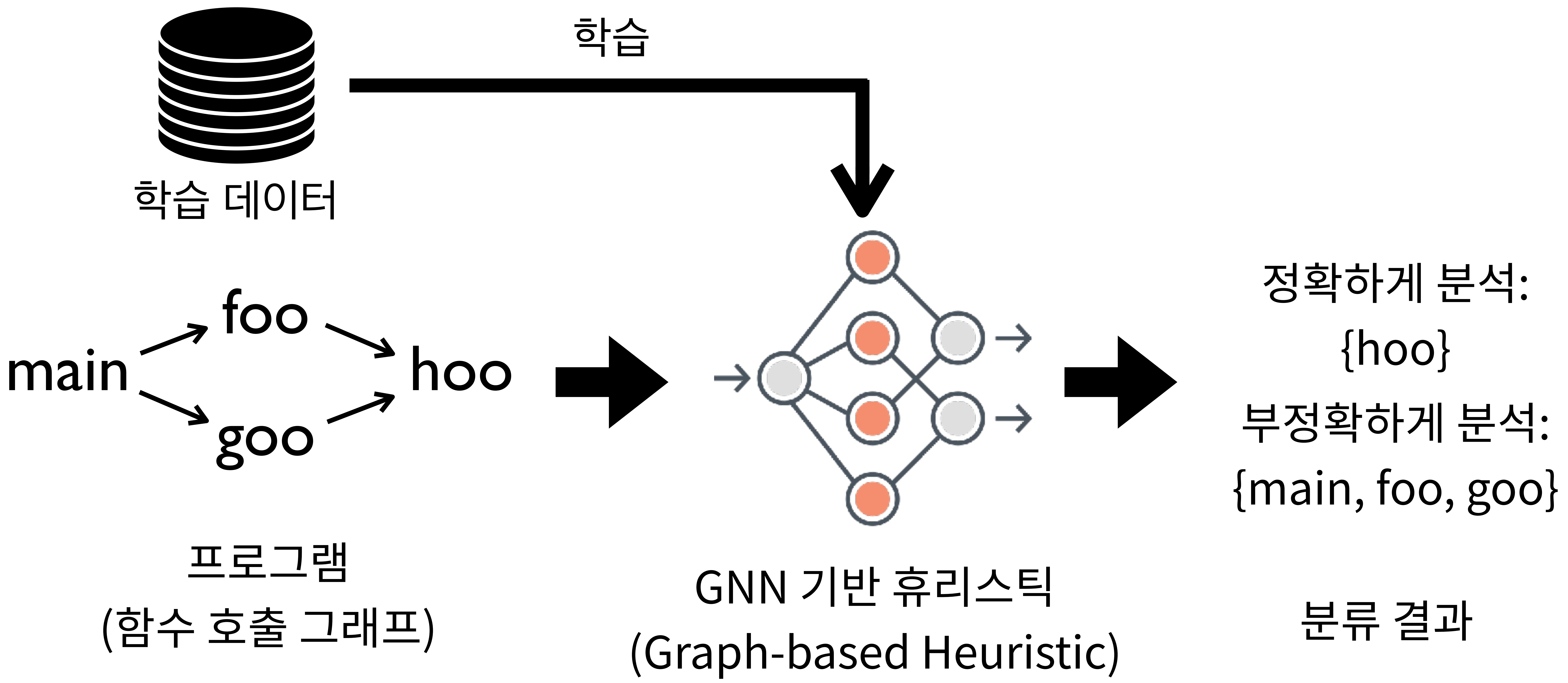
연구 시작 (2021.01)
ICML 제출 (2022.02)
NIPS 제출 (2022.05)
PLDI 제출 (2022.11)
POPL 제출 (2023.07)
PLDI 제출 (2023.11)


main
foo
goo
hoo
[2,∞], [0,0]
정확하게 분석:
{hoo}
부정확하게 분석:
{main, foo, goo}
프로그램
(함수 호출 그래프)
휴리스틱
(Graphick)
분류 결과



들어오는 엣지의 개수가 2개 이상이고 나가는 엣지의 개수가0개인 노드의 집합 (예: hoo)
main
foo
goo
hoo
[2,∞], [0,0]
들어오는 엣지
나가는 엣지
정확하게 분석:
{hoo}
부정확하게 분석:
{main, foo, goo}
프로그램
(함수 호출 그래프)
휴리스틱
(Graphick)
분류 결과


휴리스틱 합성
학습 데이터
main
foo
goo
hoo
[2,∞], [0,0]
정확하게 분석:
{hoo}
부정확하게 분석:
{main, foo, goo}
프로그램
(함수 호출 그래프)
휴리스틱
(Graphick)
분류 결과


학습
학습 데이터
main
foo
goo
hoo
프로그램
(함수 호출 그래프)
GNN 기반 휴리스틱
(Graph-based Heuristic)
정확하게 분석:
{hoo}
부정확하게 분석:
{main, foo, goo}
분류 결과

"GNNs have emerged as a cornerstone in graph learning, demonstrating exceptional performance in various applications."

- Yuan et al. [2023]

goo

부정확하게 분석:

Semi-Supervised Classification with Graph Convolutional

TN Kipf 저술 · 2016 · 36715회 인용 — Access **Paper**: View a PDF of the **paper** title

(Graph-based Heuristic)

개인적인 연구동기

- 머신러닝랩(MLV)과 합동 연구 미팅 중

낮을수록 좋은 분석

	antlr				
	recall	precision	alarms	costs	select
Minimal	-	-	463	28.12	0
Graphick	1.0	0.19	463	26.83	2629
MLP	1.0	0.108	463	91	4731
GCN	1.0	0.07	463	107.74	6183
GCN(Full)	1.0	0.069	X	X	7394
GAT	1.0	0.09	-	-	5645
GAT(Full)	1.0	0.092	-	-	5567
APPNP	1.0	0.069	-	-	7369
GCN(Concat)	1.0	0.109	-	-	4681
GCN(OnlyEdge)	1.0	0.109	-	-	4710
GCN(Primitive)	1.0	0.160	463	57.12	3194

MLV랩 학생

연구 시작 (2021.01)

ICML 제출 (2022.02)

NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

개인적인 연구동기

- 머신러닝랩(MLV)과 합동 연구 미팅 중

낮을수록 좋은 분석

	antlr				
	recall	precision	alarms	costs	select
Minimal	-	-	463	28.12	0
Graphick	1.0	0.19	463	26.83	2629
MLP	1.0	0.108	463	91	4731
GCN	1.0	0.07	463	107.74	6183
GCN(Full)	1.0	0.069	X	X	7394
GAT	1.0	0.09	-	-	5645
GAT(Full)	1.0	0.092	-	-	5567
APPNP	1.0	0.069	-	-	7369
GCN(Concat)	1.0	0.109	-	-	4681
GCN(OnlyEdge)	1.0	0.109	-	-	4710
GCN(Primitive)	1.0	0.160	463	57.12	3194

연구 시작 (2021.01) — ICML 제출 (2022.02) — NIPS 제출 (2022.05) — PLDI 제출 (2022.11) — POPL 제출 (2023.07) — PLDI 제출 (2023.11)

A New Explainable Machine Learning for Node Classification

Minseok Jeon
Software Analysis Laboratory
Korea University

29 January 2021

연구 시작 (2021.01)

ICML 제출 (2022.02)

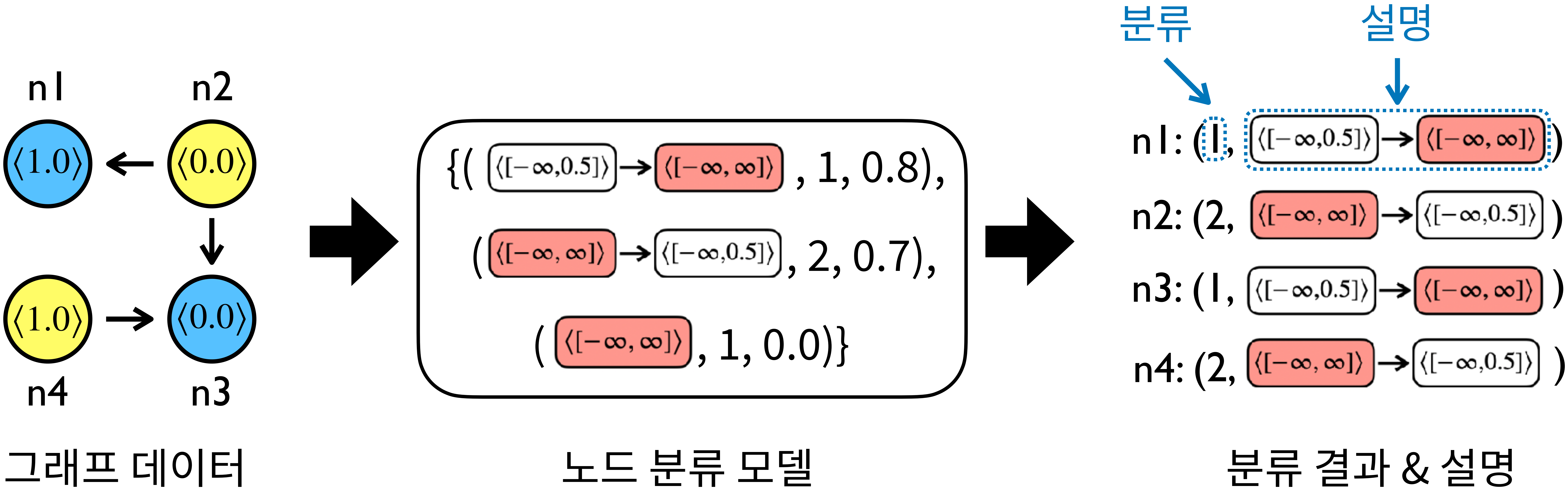
NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

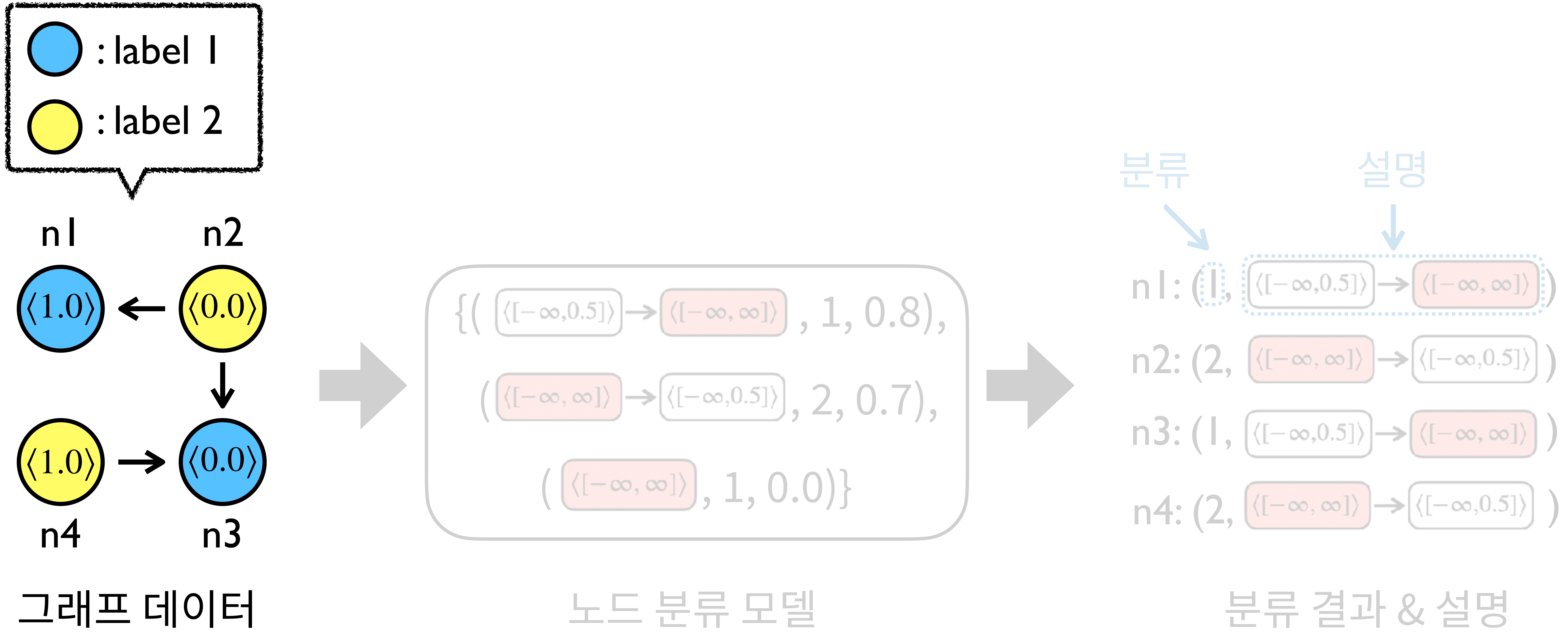
POPL 제출 (2023.07)

PLDI 제출 (2023.11)

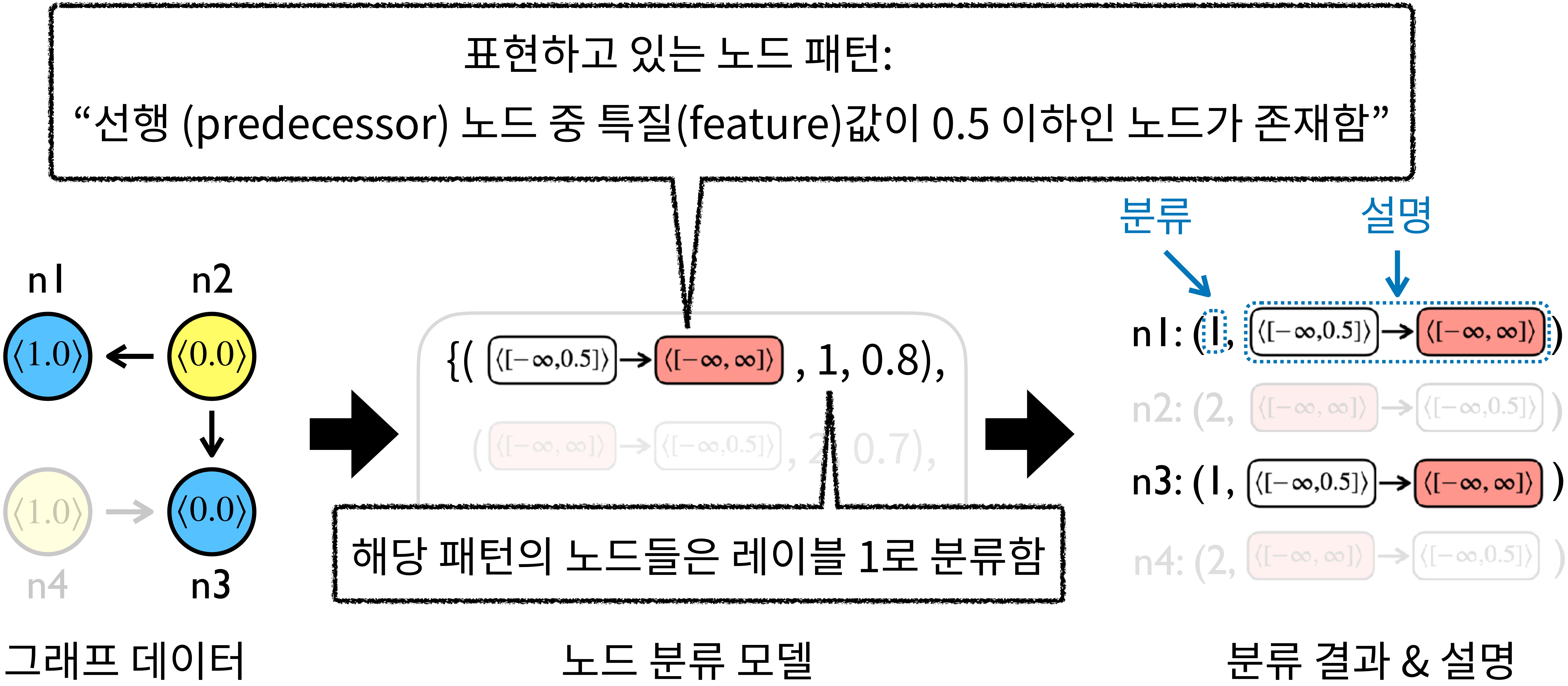

분류
설명
n1
n2
n4
n3
⟨1.0⟩ ⟨0.0⟩ ⟨1.0⟩ ⟨0.0⟩
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
n1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
그래프 데이터
노드 분류 모델
분류 결과 & 설명
연구 시작 (2021.01)
ICML 제출 (2022.02)
NIPS 제출 (2022.05)
PLDI 제출 (2022.11)
POPL 제출 (2023.07)
PLDI 제출 (2023.11)



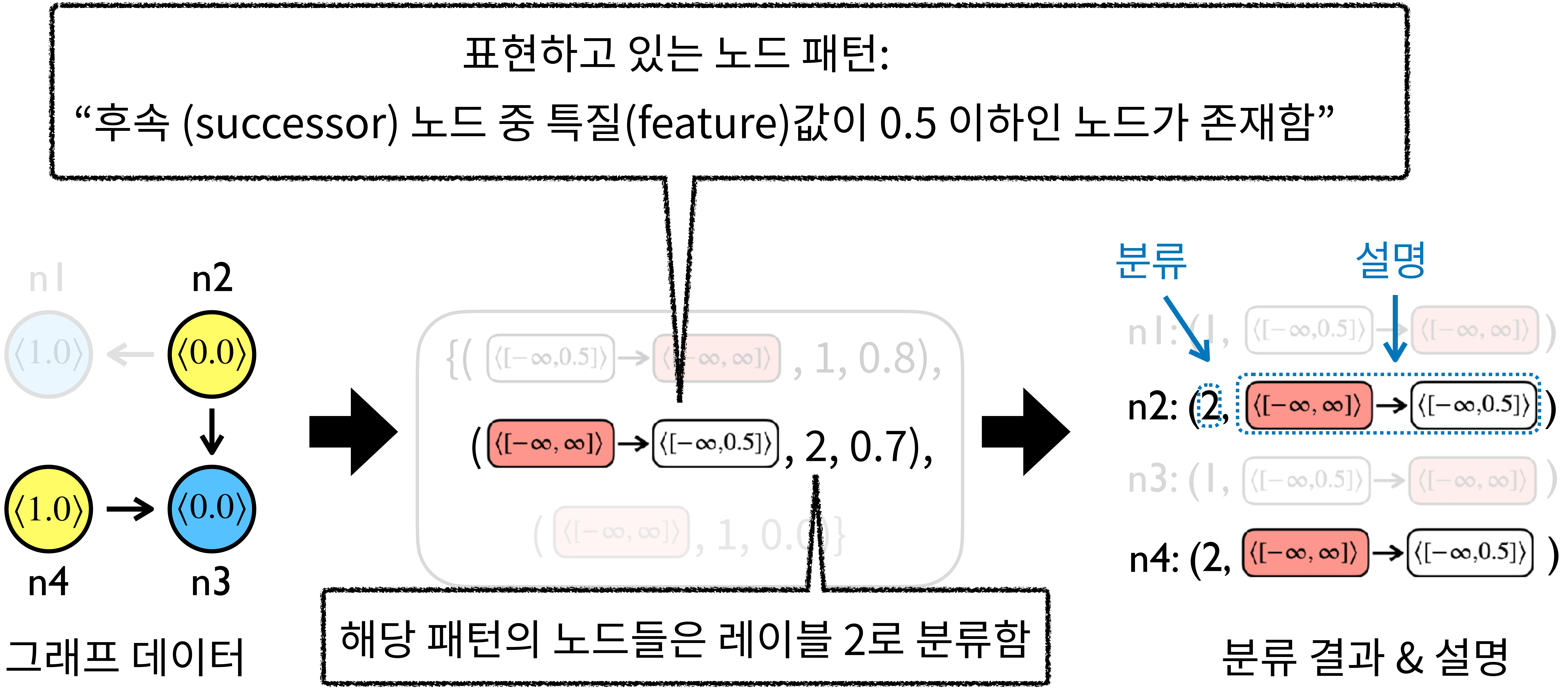
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
그래프 데이터
{(⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩, 2, 0.7),
(⟨[−∞, ∞]⟩, 1, 0.0)}
노드 분류 모델
분류
설명
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
분류 결과 & 설명



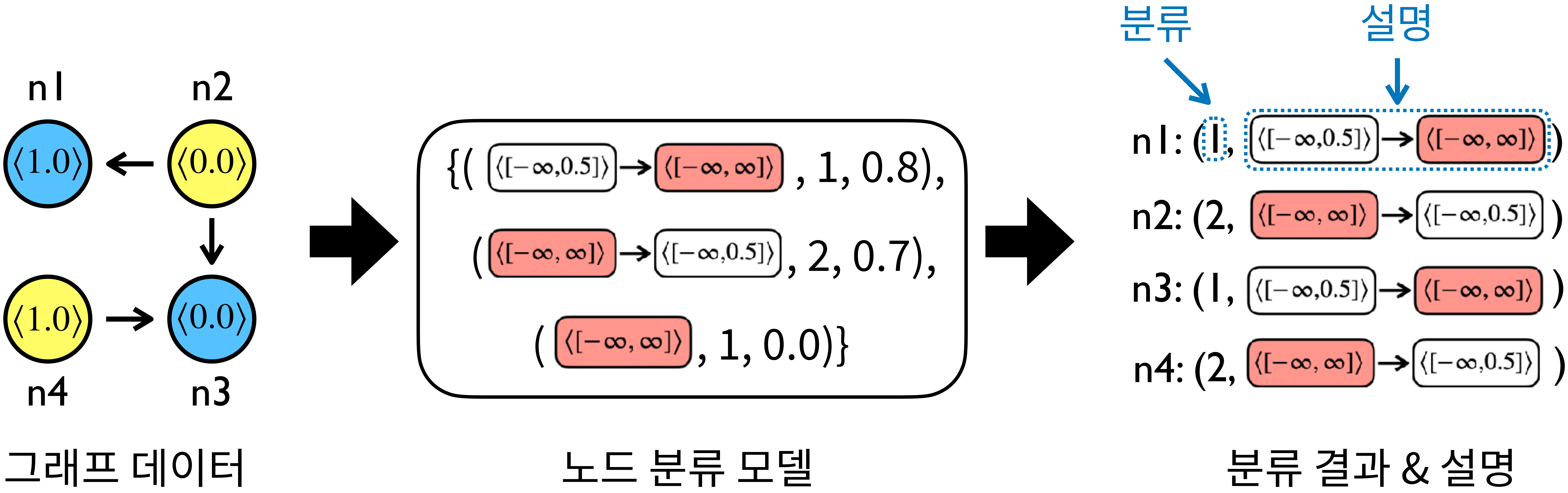
: label 1
: label 2
n1
n2
⟨1.0⟩
⟨0.0⟩
⟨1.0⟩
⟨0.0⟩
n4
n3
그래프 데이터
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩ , 2, 0.7),
(⟨[−∞, ∞]⟩ , 1, 0.0)}
노드 분류 모델
분류
설명
n1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
분류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



표현하고 있는 노드 패턴:
“선행 (predecessor) 노드 중 특질(feature)값이 0.5 이하인 노드가 존재함”
n1
n2
n3
n4
⟨1.0⟩
⟨0.0⟩
⟨0.0⟩
⟨1.0⟩
{(⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩ , 1, 0.8),
(⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩ , 2, 0.7),
해당 패턴의 노드들은 레이블 1로 분류함
분류
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”
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)}
해당 패턴의 노드들은 레이블 2로 분류함
분류
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
설명
n1: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩ → ⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩ → ⟨[−∞,0.5]⟩)
분류 결과 & 설명


모델(프로그램) 합성
학습 데이터
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
설명
n1: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n2: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
n3: (1, ⟨[−∞,0.5]⟩→⟨[−∞, ∞]⟩)
n4: (2, ⟨[−∞, ∞]⟩→⟨[−∞,0.5]⟩)
분류 결과 & 설명
연구 시작 (2021.01)
ICML 제출 (2022.02)
NIPS 제출 (2022.05)
PLDI 제출 (2022.11)
POPL 제출 (2023.07)
PLDI 제출 (2023.11)

A Formal Language-Based Model for Graph Node Classification

Anonymous Authors[1]

Abstract

We investigate a new approach to graph node classification. Our innovation, which departs significantly from dominant approaches such as Graph Neural Networks (GNNs), is that its machine-learning model consists of a formal language and therefore is interpretable by construction. To this end, our node classification technique, JARGON, is based on two ideas. First, we present a domain-specific language that can express graph structures and node features. Second, we present a learning algorithm for our model that includes sentences of the language as learnable parameters. Evaluation using widely-used datasets shows that JARGON produces simple and insightful models that are as accurate as state-of-the-art GNNs.

planations. This limitation is particularly problematic in decision-critical applications where model's transparency and interpretability are of the greatest importance (Doshi-Velez & Kim, 2017). To relieve this shortcoming, GNNs can be used with post-hoc explanation methods (Ying et al., 2019; Luo et al., 2020; Vu & Thai, 2020; Yuan et al., 2021), but explaining black-box GNN models by a separate process is fundamentally challenging and several problems remain unsolved until recently (Yuan et al., 2020b).

This Work. In this paper, we explore a radically different approach to machine learning on graphs. The most distinctive feature of our approach, which departs significantly from the dominant GNN approaches, is the use of a formal language to describe graphs, which allows our model to be inherently interpretable without ambiguity. Yet, our model can make accurate predictions as the language is expressive enough to capture complex structural properties of graphs.

Evaluation

- 정확도 비교

Model	BA-SHAPES	TREE-CYCLES
GCN	0.957	0.977
GAT	0.900	0.981
JARGON	**0.971**	**1.0**

Model	CORA	CITESEER	PUBMED
GCN	**0.886**	0.764	**0.882**
GAT	0.874	0.766	0.868
JARGON	0.882	**0.780**	**0.882**

연구 시작 (2021.01)

ICML 제출 (2022.02)

NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

A Formal Language-Based Model for Graph Node Classification

점수 : Reject X 2, Weak accept X 2

"The major concern is the experimental study. **Only two weak baselines, GCN and GAT**, are used to compare the node classification performance."

최종 결과 : Reject

JARGON produces simple and insightful models that are as accurate as state-of-the-art GNNs.

language to describe graphs, which allows our model to be inherently interpretable without ambiguity. Yet, our model can make accurate predictions as the language is expressive enough to capture complex structural properties of graphs.

Evaluation

- 정확도 비교

Model	BA-SHAPES	TREE-CYCLES
GCN	0.957	0.977
GAT	0.900	0.981
JARGON	**0.971**	**1.0**

Model	CORA	CITESEER	PUBMED
GCN	**0.886**	0.764	**0.882**
GAT	0.874	0.766	0.868
JARGON	0.882	**0.780**	**0.882**

연구 시작 (2021.01)
ICML 제출 (2022.02)
NIPS 제출 (2022.05)
PLDI 제출 (2022.11)
POPL 제출 (2023.07)
PLDI 제출 (2023.11)


제출
떨어짐
개선
연구 시작
(2021.01)
ICML 제출
(2022.02)
NIPS 제출
(2022.05)
PLDI 제출
(2022.11)
POPL 제출
(2023.07)
PLDI 제출
(2023.11)

벤치마크 보강

베이스라인 추가

	CORA	CITESEER	PUBMED	WISCONSIN	TEXAS	CORNELL	BA-SHAPES	TREE-CYCLES
GCN	86.9±0.4	76.7±0.3	85.5±0.1	64.0±1.5	69.9±4.3	61.1±0.0	95.7±0.0	97.7±0.0
GAT	86.4±0.8	76.4±0.3	83.5±0.1	58.4±3.5	63.3±7.3	66.5±3.0	89.6±3.7	99.0±1.1
CHEBYNET	85.0±0.7	74.8±0.5	84.3±1.5	69.6±3.9	69.9±2.6	59.9±7.6	89.3±8.1	82.4±17.8
APPNP	85.9±0.9	75.9±0.2	86.5±0.1	43.2±2.9	57.7±2.6	63.2±9.3	87.1±1.7	91.3±0.5
JKNET	86.8±0.5	**76.9±0.2**	87.4±0.2	60.8±1.5	82.1±6.3	61.1±0.0	95.7±0.0	97.8±0.7
GRAPHSAGE	87.0±0.2	76.8±0.4	**87.6±0.1**	84.0±0.0	**89.9±4.0**	**88.8±6.9**	96.5±0.6	**100.0±0.0**
JARGON	**87.7±0.0**	76.8±0.0	83.6±0.0	**88.0±0.0**	83.3±0.0	77.7±0.0	**97.1±0.0**	**100.0±0.0**

연구 시작 (2021.01)

ICML 제출 (2022.02)

NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

A Formal Language-Based Model for Graph Node Classification

Anonymous Author(s)
Affiliation
Address
email

Abstract

We investigate a new approach to graph node classification. Our innovation, which departs significantly from dominant approaches such as Graph Neural Networks (GNNs), is that its machine-learning model consists of a formal language and therefore is interpretable by construction. To this end, our node classification technique, called JARGON, works with two ideas. First, we present a domain-specific language that can express graph structures and node features. Second, we offer a learning algorithm for our model that includes sentences of the language as learnable parameters. Evaluation using widely-used datasets shows that JARGON produces simple and insightful models that are as accurate as representative GNNs.

점수 :

Weak reject X 2, Weak accept X 2

연구 시작 (2021.01)

ICML 제출 (2022.02)

NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

A Formal Language-Based Model for Graph Node Classification

Anonymous Author(s)
Affiliation
Address
email

Abstract

We investigate a new approach to graph node classification. Our innovation, which departs significantly from dominant approaches such as Graph Neural Networks (GNNs), is that its machine-learning model consists of a formal language and therefore is interpretable by construction. To this end, our node classification technique, called JARGON, works with two ideas. First, we present a domain-specific language that can express graph structures and node features. Second, we offer a learning algorithm for our model that includes sentences of the language as learnable parameters. Evaluation using widely-used datasets shows that JARGON produces simple and insightful models that are as accurate as representative GNNs.

점수 :
Weak reject X 2, Weak accept X 2

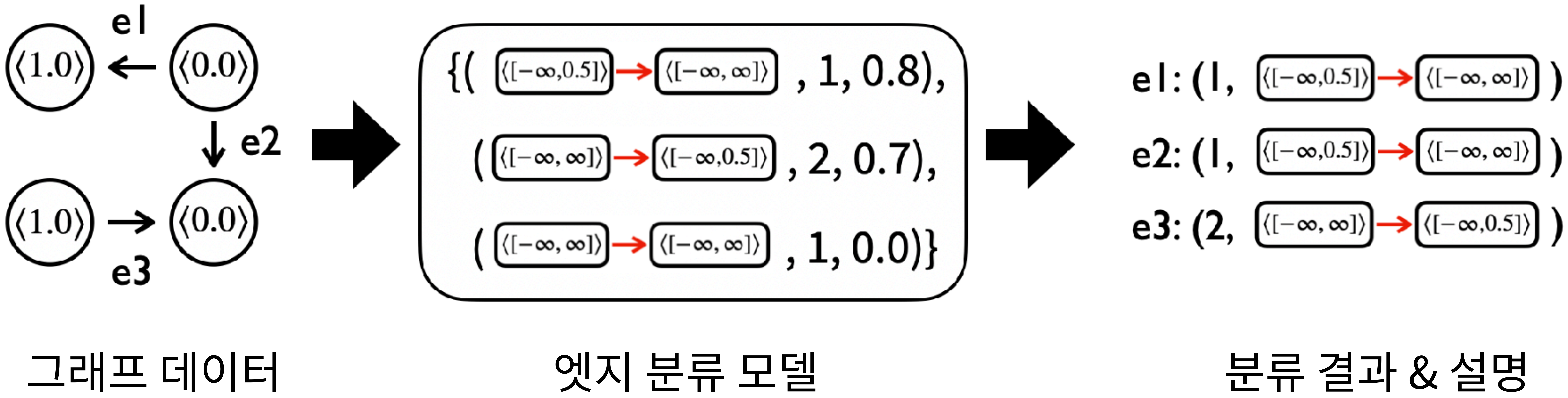
"How easy is it to design a different domain specific language for **other graph problems such as link-prediction**?"

최종 결과 : Rejected

연구 시작 (2021.01)
ICML 제출 (2022.02)
NIPS 제출 (2022.05)
PLDI 제출 (2022.11)
POPL 제출 (2023.07)
PLDI 제출 (2023.11)


제출
떨어짐
개선
연구 시작
(2021.01)
ICML 제출
(2022.02)
NIPS 제출
(2022.05)
PLDI 제출
(2022.11)
POPL 제출
(2023.07)
PLDI 제출
(2023.11)

(1) 엣지 분류가 가능하도록 확장



연구 시작 (2021.01)

ICML 제출 (2022.02)

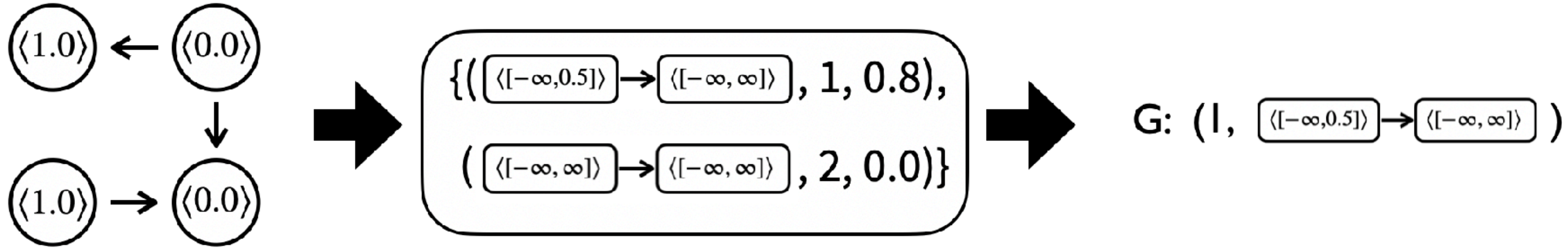
NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

(2) 그래프 분류가 가능하도록 확장



그래프 데이터 G

그래프 분류 모델

분류 결과 & 설명

연구 시작 (2021.01)

ICML 제출 (2022.02)

NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

A Programming Language Approach to Graph Learning

ANONYMOUS AUTHOR(S)

In this article, we present a novel, language-based approach to graph learning. The main feature, which departs significantly from the dominant approach based on Graph Neural Networks (GNNs), is that our machine-learning model consists of a formal language and is therefore interpretable by construction. Our approach, called JARGON, is built on two techniques widely known in the programming languages community. First, we use abstract interpretation to design a language describing abstract graphs whose semantics is a set of concrete graphs; "executing" an abstract graph performs classification based on the concrete graphs that it denotes. Second, we cast learning as a program synthesis problem, and present top-down and bottom-up algorithms for learning abstract graphs from training data. Evaluation using widely-used datasets shows that JARGON produces models that are simple and insightful, yet the learned models can compete with representative GNNs. For the real-world MUTAG dataset for graph classification, for example, our learning algorithm produced a small model with 22 easy-to-interpret abstract graphs while achieving a classification accuracy of 95% on unseen data, outperforming well-known GNN models such as GIN (Graph Isomorphism Network).

CCS Concepts: • **Software and its engineering** → *Domain specific languages*; • **Computing methodologies** → *Supervised learning*.

Additional Key Words and Phrases: Graph Learning, Formal Language, Program Synthesis

ACM Reference Format:
Anonymous Author(s). 2018. A Programming Language Approach to Graph Learning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference 'XX)*. ACM, New York, NY, USA, 24 pages. https://doi.org/XXXXXXX.XXXXXXX

"We consider three types of classification tasks on graphs: node, edge, graph classification."

"In the graph classification dataset MUTAG, Jargon shows the best accuracy"

연구 시작 (2021.01)

ICML 제출 (2022.02)

NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

점수

Review #534A	Reject
Review #534B	Weak Reject
Review #534C	Weak Reject
Review #534D	Weak accept

최종 결과 : Reject

- 문제 1: 설명에 대한 정량적 비교 부족
- 문제 2: 너무 비싼 학습 비용
- 문제 3: 프로그래밍 언어 분야와의 연관성

연구 시작 (2021.01)

ICML 제출 (2022.02)

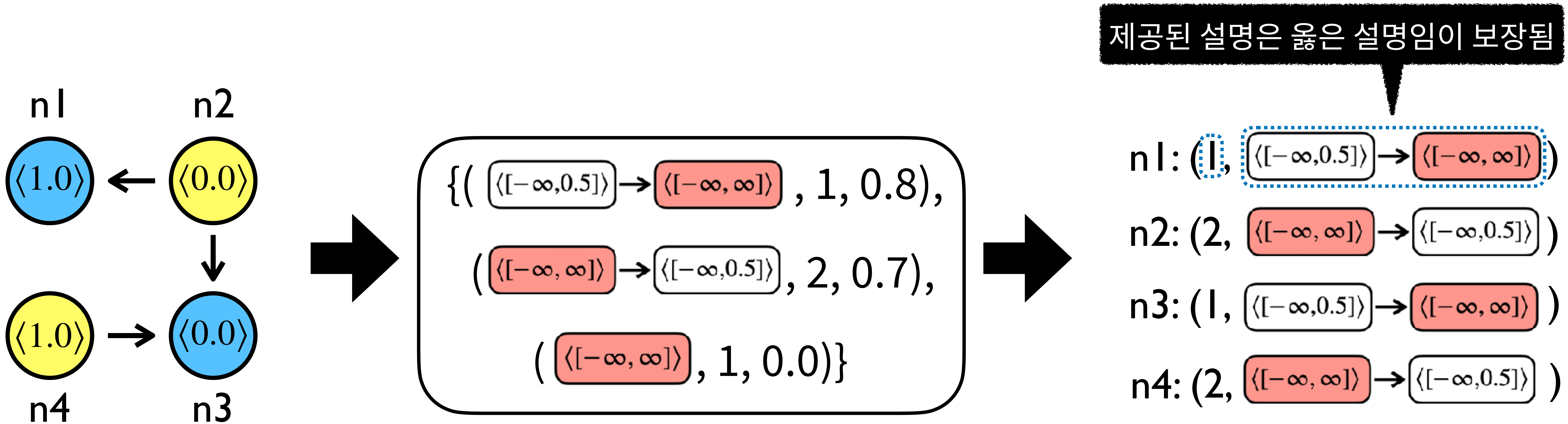
NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

문제 1: 설명에 대한 정량적 비교 부족



"The authors need to come up with a systematic, head-to-head comparison with a well-defined metric to measure the explainability."

문제 2: 비싼 학습 비용

- 학습 비용 비교(분)

	MUTAG	BBBP	BACE	Cora	Citeseer	Pubmed	Texas	Cornell	Wisconsin
GNN	0.2	1.0	1.0	0.4	0.4	0.6	0.4	0.3	0.4
Ours	12.3	34.3	60.6	61.6	245.2	2702.9	5.0	5.0	8.0
	61x↑	34x↑	60x↑	154x↑	613x↑	4504x↑	12x↑	16x↑	20x↑

"The main concern I have is the scalability of the approach. Training is too expensive."

"I fear that scalability will inherently be a problem with the current approach."

문제 3: 프로그래밍 언어와의 연관성

- 핵심 기술: DSL (요약 그래프)

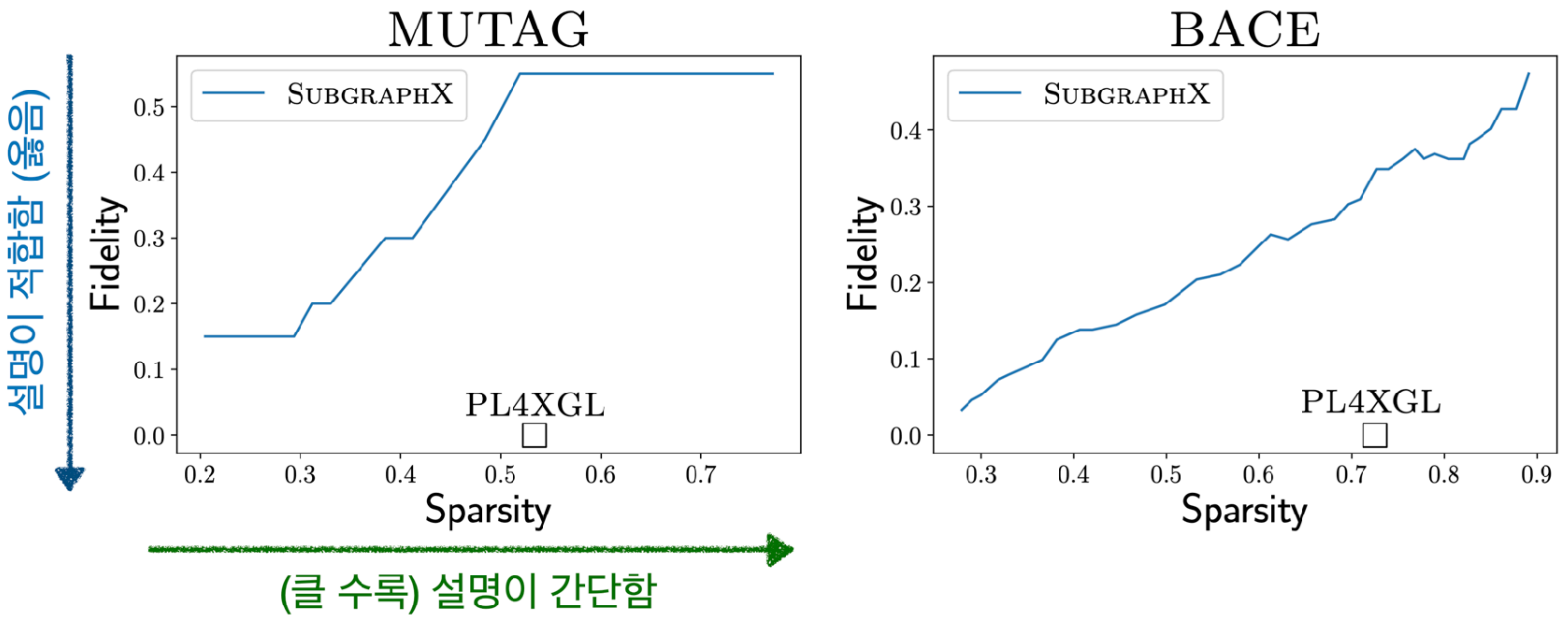
$$\boxed{\langle[-\infty,0.5]\rangle} \rightarrow \boxed{\langle[-\infty,\infty]\rangle} \in \widehat{Graph}$$

$$\begin{aligned}
\widehat{Graph} &= \widehat{Node}^{*} \times \widehat{Edge}^{*} \\
\widehat{Node} &= Itv^{n} \\
\widehat{Edge} &= \mathbb{N} \times \mathbb{N} \times Itv^{m}
\end{aligned}$$

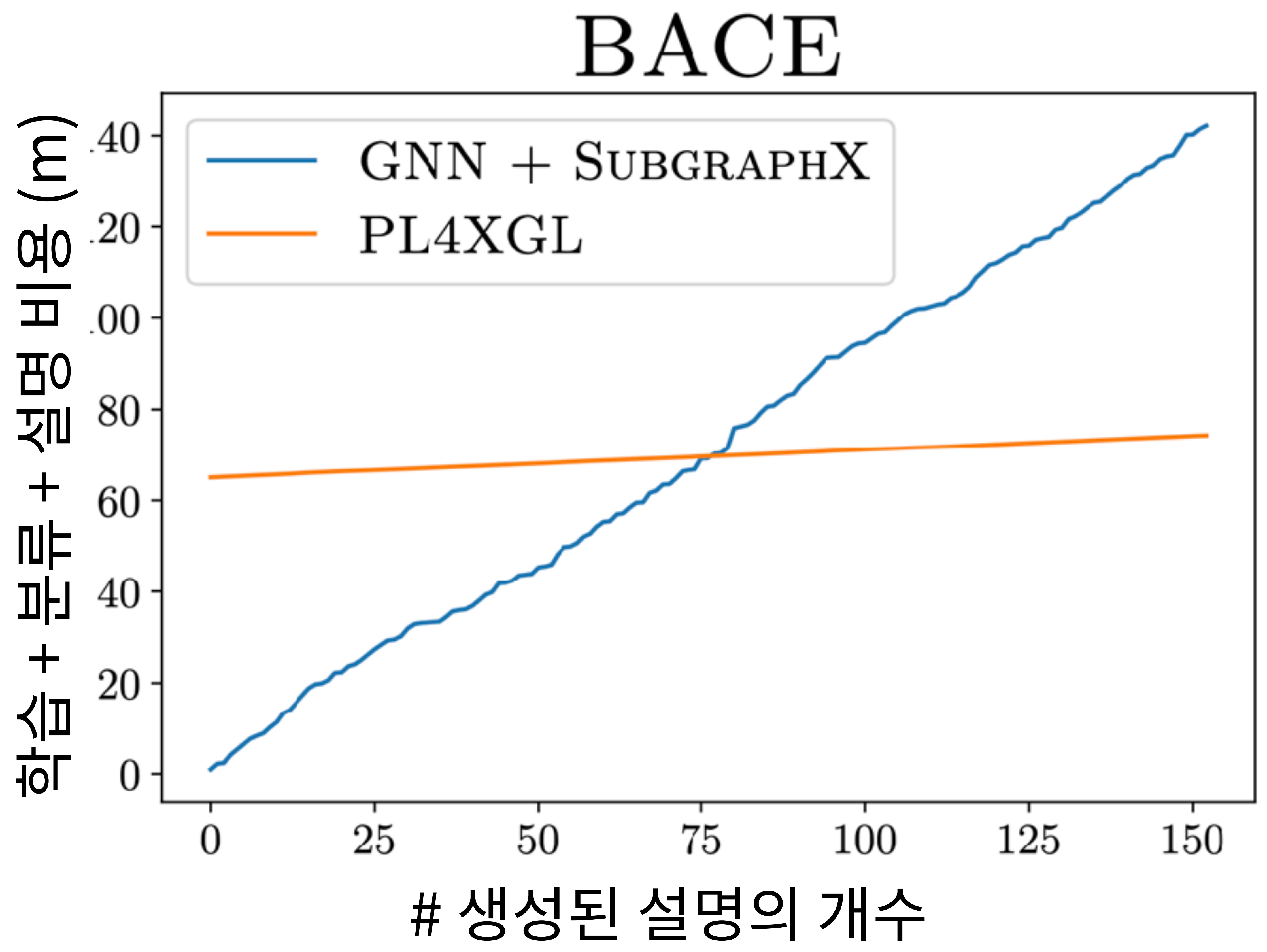
“**Relation to Programming Languages.**
I had a hard time trying to relate abstract graphs to a DSL.”

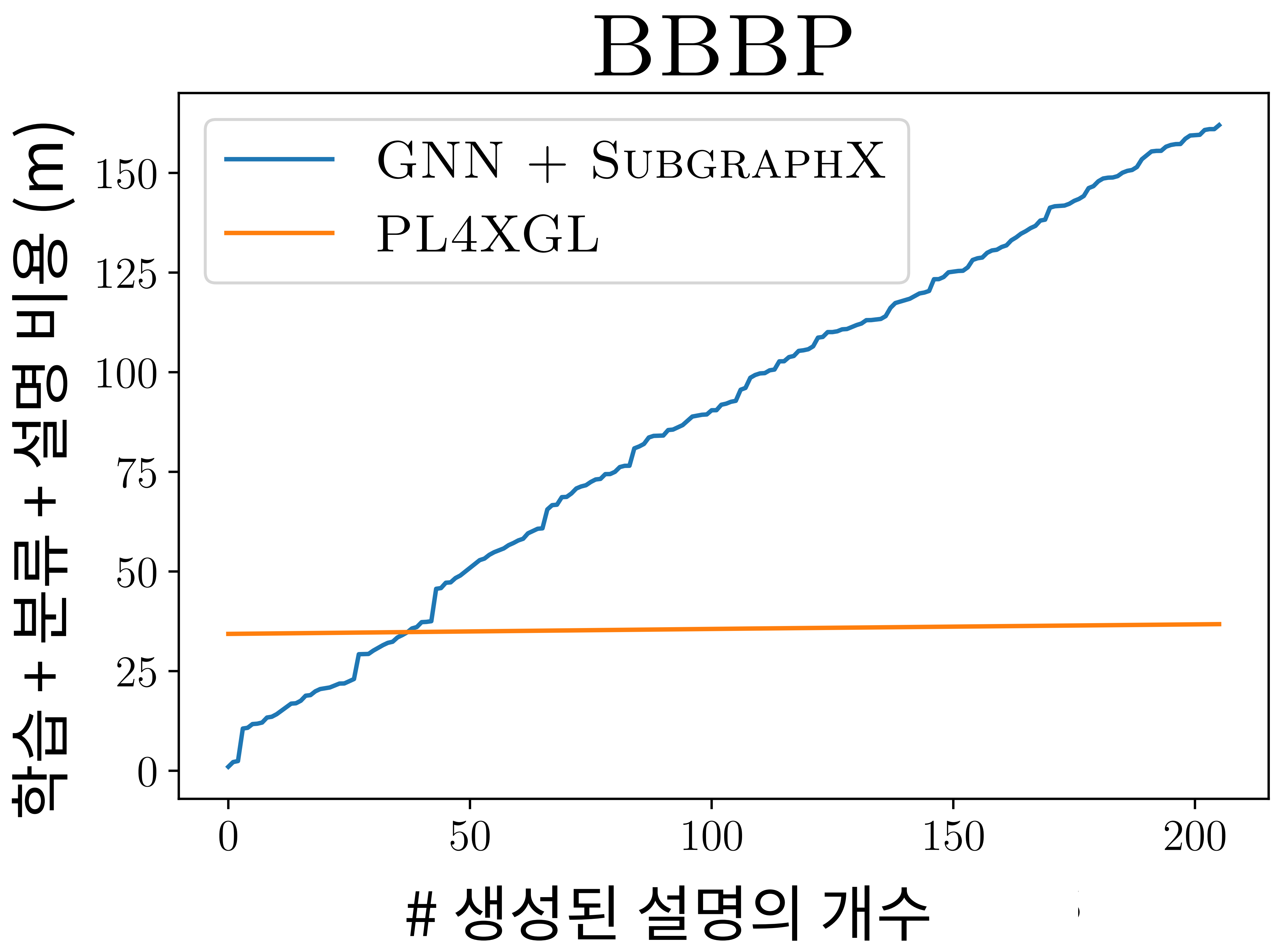

제출
떨어짐
개선
연구 시작
(2021.01)
ICML 제출
(2022.02)
NIPS 제출
(2022.05)
PLDI 제출
(2022.11)
POPL 제출
(2023.07)
PLDI 제출
(2023.11)

(1) 설명의 정량적 비교 실험 추가



(2) 총 비용(학습+분류+설명)으로 비교하기





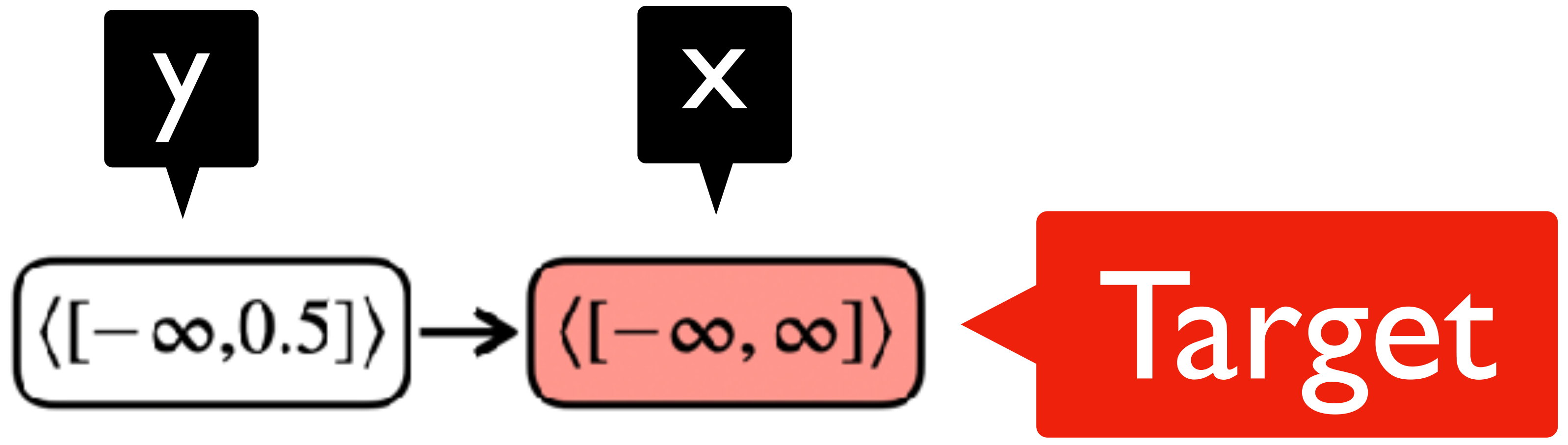
연구 시작	ICML 제출	NIPS 제출	PLDI 제출	POPL 제출	PLDI 제출
(2021.01)	(2022.02)	(2022.05)	(2022.11)	(2023.07)	(2023.11)

(3) 요약 그래프를 기반으로 도메인 특화 프로그래밍 언어 GDL(Graph Description Language) 디자인

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

GDL 프로그램

표현하는 패턴



요약 그래프

연구 시작 (2021.01)
ICML 제출 (2022.02)
NIPS 제출 (2022.05)
PLDI 제출 (2022.11)
POPL 제출 (2023.07)
PLDI 제출 (2023.11)

(3) 요약 그래프를 기반으로 도메인 특화 프로그래밍 언어 GDL(Graph Description Language) 디자인

$$\widehat{Graph} = \widehat{Node}^* \times \widehat{Edge}^*$$
$$\widehat{Node} = Itv^n$$
$$\widehat{Edge} = \mathbb{N} \times \mathbb{N} \times Itv^m$$

요약 그래프

박지혁 교수님

Programs	P_4	::= $\overline{\delta}$ **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 | ...

GDL 문법

연구 시작 (2021.01) — ICML 제출 (2022.02) — NIPS 제출 (2022.05) — PLDI 제출 (2022.11) — POPL 제출 (2023.07) — PLDI 제출 (2023.11)

PL4XGL: A Programming Language Approach to Explainable Graph Learning

ANONYMOUS AUTHOR(S)

In this article, we present a new, language-based approach to explainable graph learning. Though graph neural networks (GNNs) have shown impressive performance in various graph learning tasks, they have severe limitations in explainability, hindering their use in decision-critical applications. Recently, several GNN explanation techniques have been proposed using a *post-hoc* explanation approach with *subgraphs* as explanations for classification results. Unfortunately, however, they have fundamental drawbacks in terms of 1) *additional cost*, 2) *correctness*, and 3) *generality* of explanations. This paper aims to address these problems by developing a new graph-learning method based on programming language techniques. Our key idea is two-fold: 1) designing a *graph description language (GDL)* to explain the classification results instead of subgraphs and 2) developing a new *GDL-based interpretable classification model* instead of GNN-based models. Our graph-learning model, called PL4XGL, consists of a set of candidate GDL programs with labels and quality scores. For a given graph component, it searches the best applicable GDL program and provides the corresponding label as the classification result and the program as the explanation. In our approach, learning from data is formulated as a program-synthesis problem, and we present top-down and bottom-up algorithms for synthesizing GDL programs from training data. Evaluation using widely-used datasets demonstrates that PL4XGL produces high-quality explanations that outperform those produced by the state-of-the-art GNN explanation technique, SUBGRAPHX. Furthermore, we show that PL4XGL has more accurate classification results with an endurable learning cost than popular GNN models.

CCS Concepts: • **Software and its engineering** → *Domain specific languages*; • **Computing methodologies** → *Supervised learning*.

Additional Key Words and Phrases: Graph Learning, Domain-Specific Language, Program Synthesis

ACM Reference Format:
Anonymous Author(s). 2018. PL4XGL: A Programming Language Approach to Explainable Graph Learning. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference 'XX)*. ACM, New York, NY, USA, 29 pages. https://doi.org/XXXXXXX.XXXXXXX

"We design a graph description language, called GDL, as a declarative programming language in which a program describes a set of nodes, edges, or graphs."

"PL4XGL outperforms SubgraphX in terms of *Fidelity* for all datasets, achieving the optimal score of 0.0."

"PL4XGL eventually **outperforms** the baseline in terms of the accumulated (training + classification + explanation) cost."

연구 시작 (2021.01)

ICML 제출 (2022.02)

NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

점수

Review #958A	**Weak reject**
Review #958B	**Strong accept**
Review #958C	**Weak reject**

연구 시작 (2021.01) | ICML 제출 (2022.02) | NIPS 제출 (2022.05) | PLDI 제출 (2022.11) | POPL 제출 (2023.07) | PLDI 제출 (2023.11)

점수

Review #958A	Weak reject
Review #958B	Strong accept
Review #958C	Weak reject

최종 결과 : Reject

"It would have been very helpful to have included the implementation in the initial submission."

PL4XGL: A Programming Language Approach to Explainable Graph Learning

ANONYMOUS AUTHOR(S)

In this article, we present a new, language-based approach to explainable graph learning. Though graph neural networks (GNNs) have shown impressive performance in various graph learning tasks, they have severe limitations in explainability, hindering their use in decision-critical applications. To address these limitations, several GNN explanation techniques have been proposed using a post-hoc explanation approach providing subgraphs as explanations for classification results. Unfortunately, however, they have two fundamental drawbacks in terms of 1) additional explanation costs and 2) the correctness of the explanations. This paper aims to address these problems by developing a new graph-learning method based on programming language techniques. Our key idea is two-fold: 1) designing a graph description language (GDL) to explain the classification results and 2) developing a new GDL-based interpretable classification model instead of GNN-based models. Our graph-learning model, called PL4XGL, consists of a set of candidate GDL programs with labels and quality scores. For a given graph component, it searches the best GDL program describing the component and provides the corresponding label as the classification result and the program as the explanation. In our approach, learning from data is formulated as a program-synthesis problem, and we present top-down and bottom-up algorithms for synthesizing GDL programs from training data. Evaluation using widely-used datasets demonstrates that PL4XGL produces high-quality explanations that outperform those produced by the state-of-the-art GNN explanation technique, SubgraphX. We also show that PL4XGL achieves competitive classification accuracy comparable to popular GNN models.

연구 시작 (2021.01)

ICML 제출 (2022.02)

NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

PLDI 제출 (2023.11)

점수

Review #875A	**Accept**
Review #875B	**Weak accept**
Review #875C	**Weak accept**
Review #875D	**Accept**

"PL4XGL is a nice application of synthesis algorithm for graph learning and explanation. Evaluation is also in favor. The paper would be a nice addition to the community. Thanks for the great work!"

연구 시작 (2021.01)

ICML 제출 (2022.02)

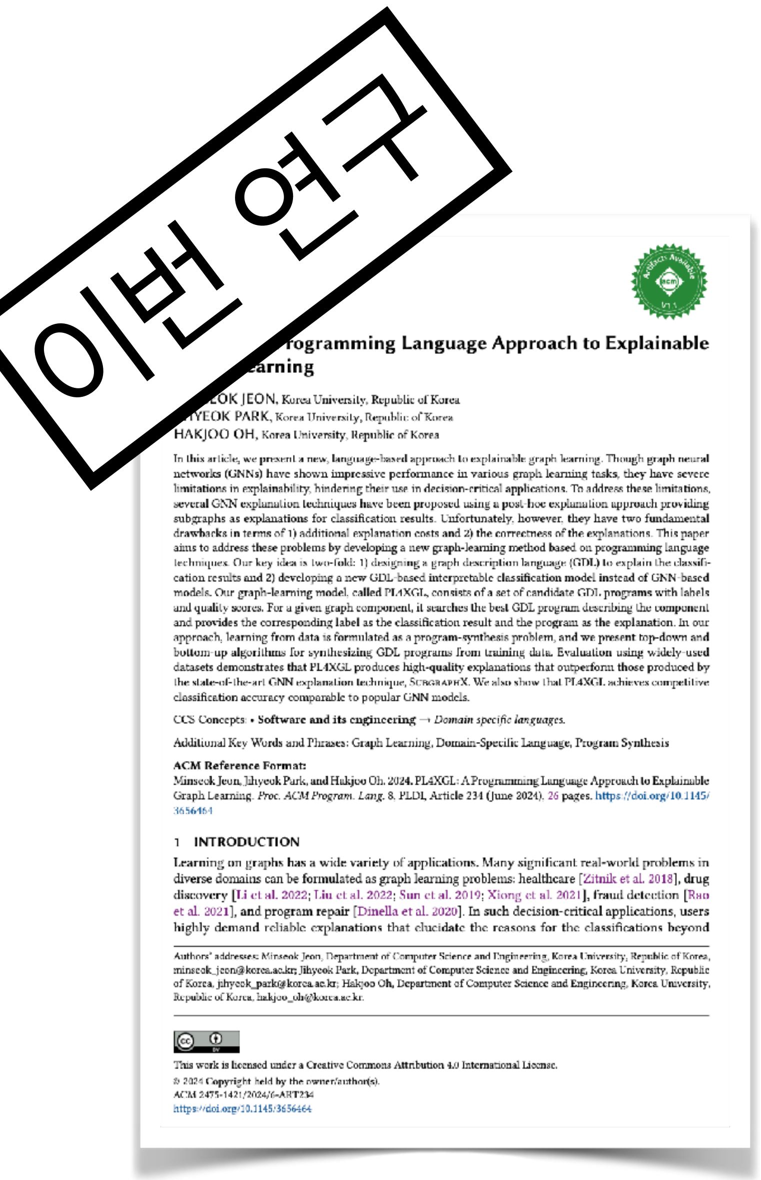
NIPS 제출 (2022.05)

PLDI 제출 (2022.11)

POPL 제출 (2023.07)

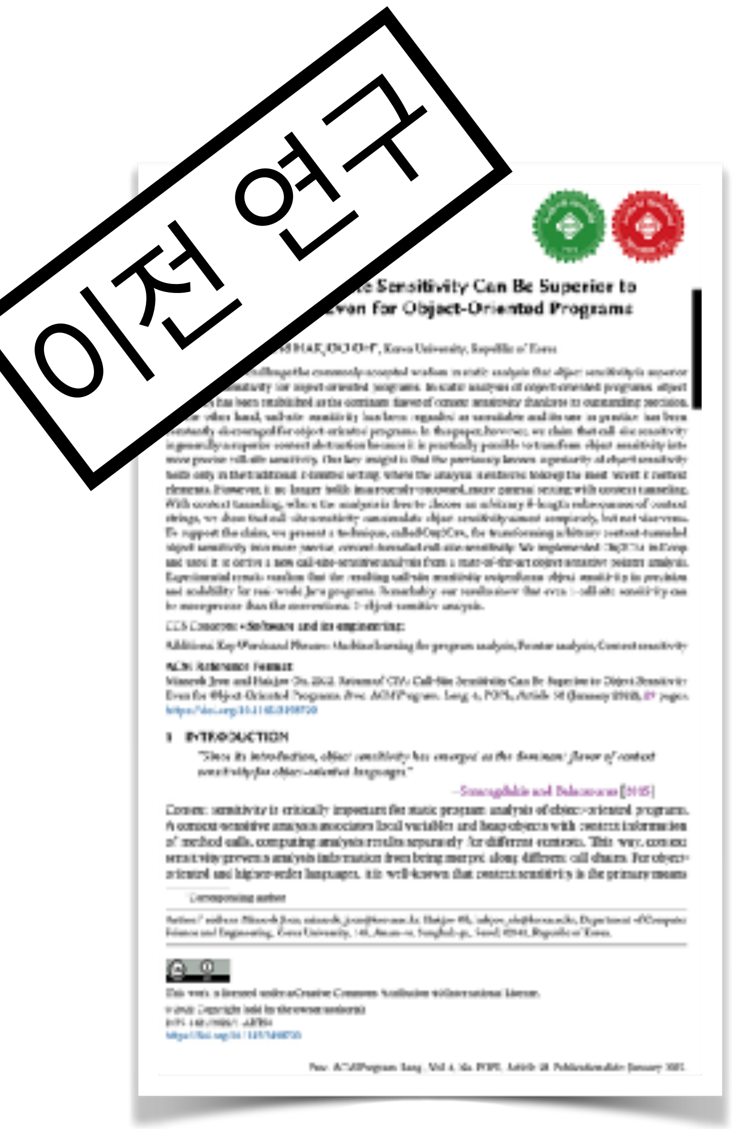
PLDI 제출 (2023.11)

될 때까지 개선하기



- 연구기간: 3년 (2021.01~2023.11)

ICML2022: Rejected → (개선) → NIPS2022: Rejected → (개선) → PLDI2023: Rejected → (개선) → POPL2024: Rejected → (개선) → PLDI2024: Accepted



- 연구기간: 3년 (2018.11~2022.01)

OOPSLA19: Rejected → (개선) → PLDI2020: Rejected → (개선) → ICSE2021: Rejected → (개선) → OOPSLA21: Rejected → (개선) → POPL2022: Accepted

OOPSLA' 20

데이터 기반 정적 분석을
위한 피처 자동 생성

Submitted

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

GDL을 사용하여
GNN 개선하기

ToDo

GDL 기반
그래프 데이터 마이닝

ToDo

GDL 기반
GNN 설명 기법

감사합니다!