Improving ML-based Binary Function Similarity Detection by Assessing and Deprioritizing Control Flow Graph Features

Jialai Wang*, Chao Zhang*‡§, Longfei Chen*, Yi Rong*, Yuxiao Wu†, Hao Wang*, Wende Tan*, Qi Li*, and Zongpeng Li*♣§

*Tsinghua University, †Huazhong University of Science and Technology, ‡Zhongguancun Laboratory, ♣Quancheng Labs

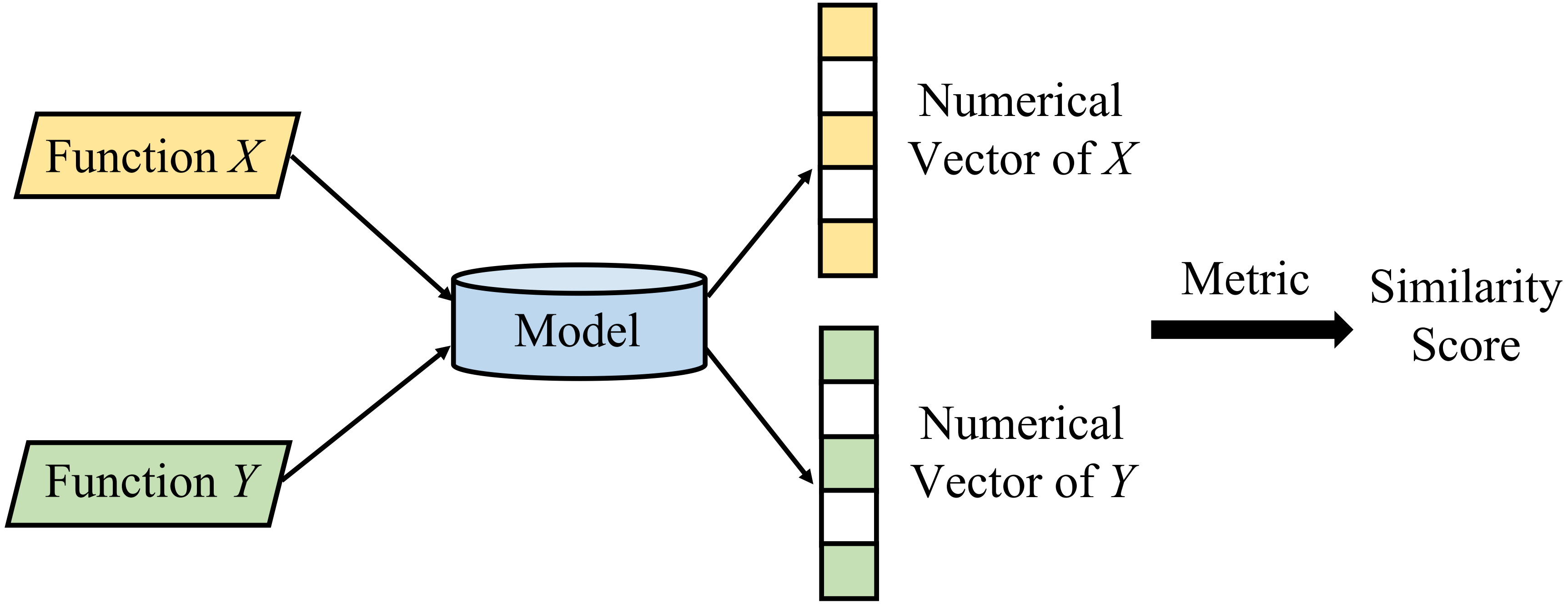
{wang-jl22, clf23, rongy19, hao-wang20}@mails.tsinghua.edu.cn, wyxhustcse@hust.edu.cn, twd2.me@gmail.com, {chaoz, qli01, zongpeng}@tsinghua.edu.cn

➢ **Key Contributions:**

- Assess the role of CFG features in ML-BFSD models
- Apply explanation methods to ML-BFSD models and reveal heavy reliance on CFG features in existing models
- Propose CFG manipulation solution (δCFG)
- Improve performance of existing models

Background

- ML-based binary function similarity detection (ML-BFSD):



Patch Analysis

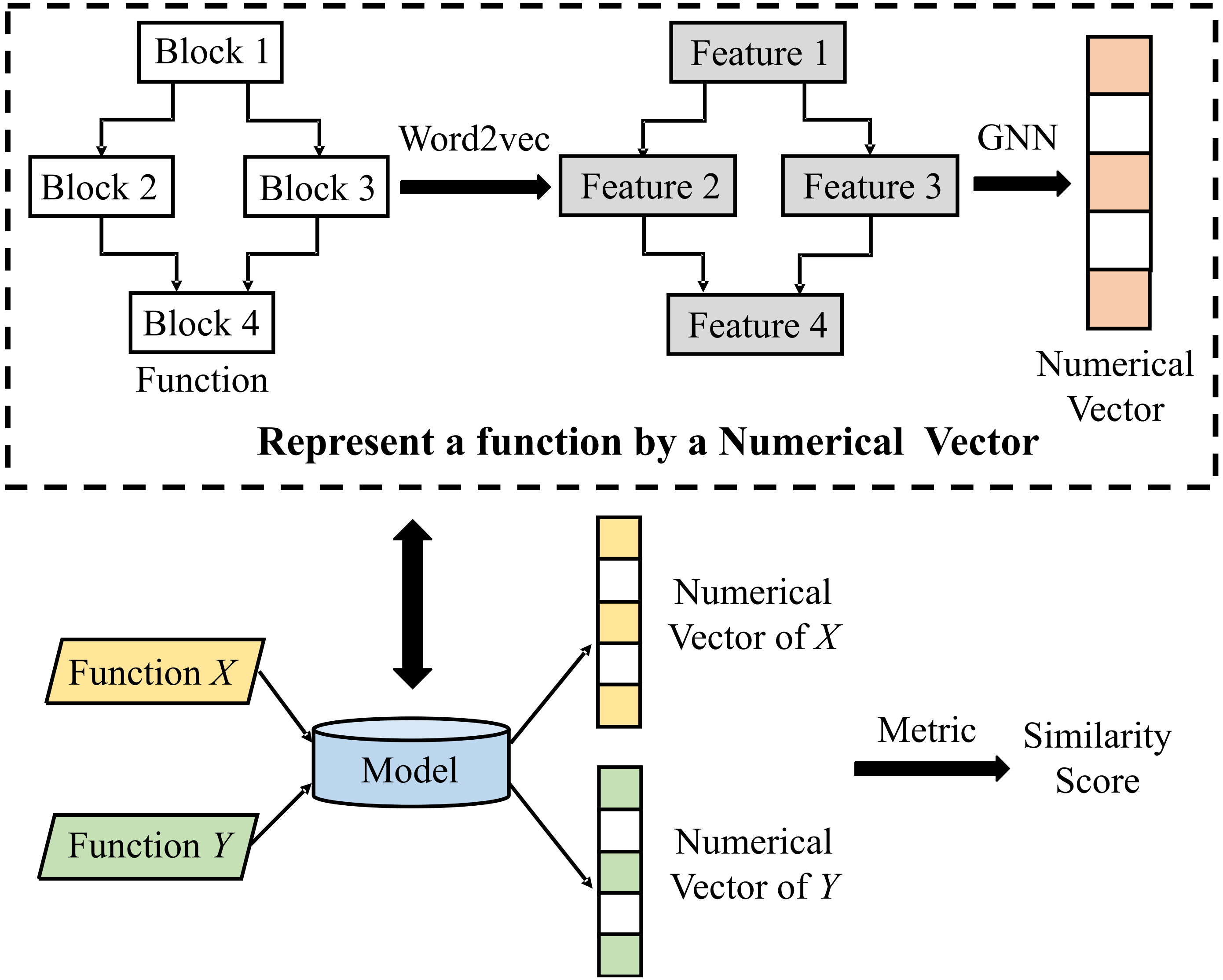
Software Plagiarism Detection

Vulnerability Discovery

ML-BFSD solutions have been widely used

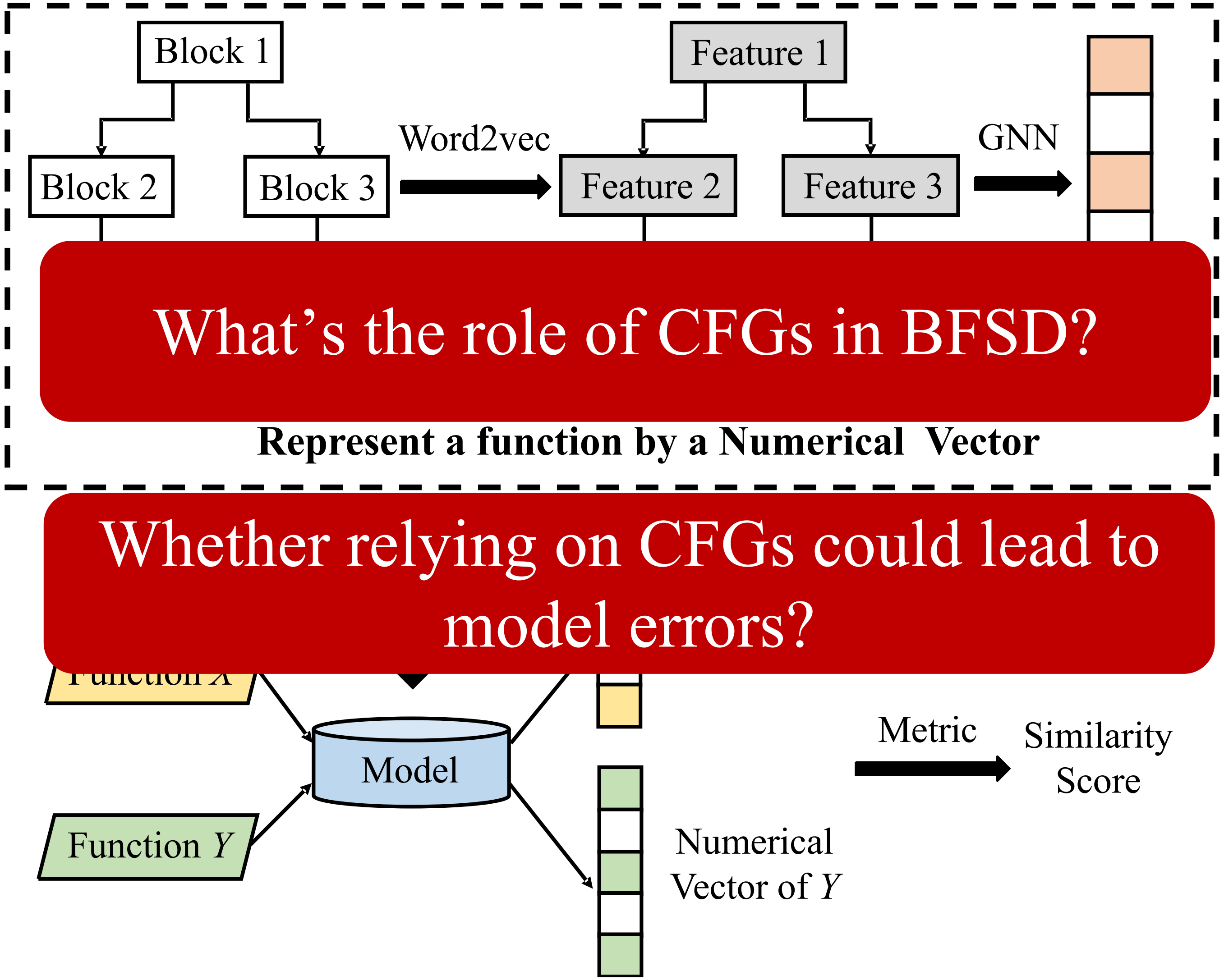
Intuition

- CFGs are important in ML-BFSD:



Intuition

- CFGs are important in ML-BFSD:

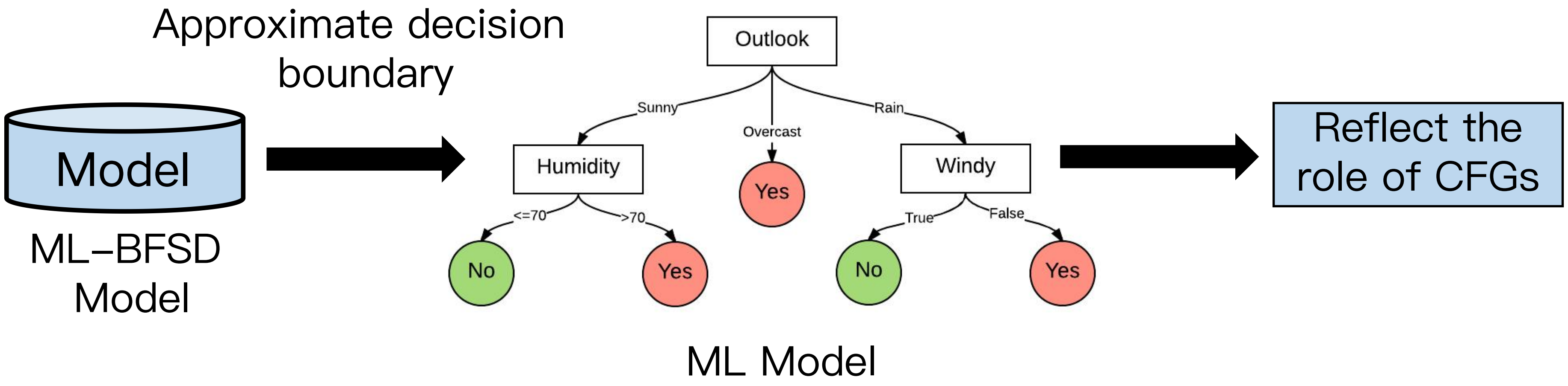


What's the role of CFGs in BFSD?

Whether relying on CFGs could lead to model errors?

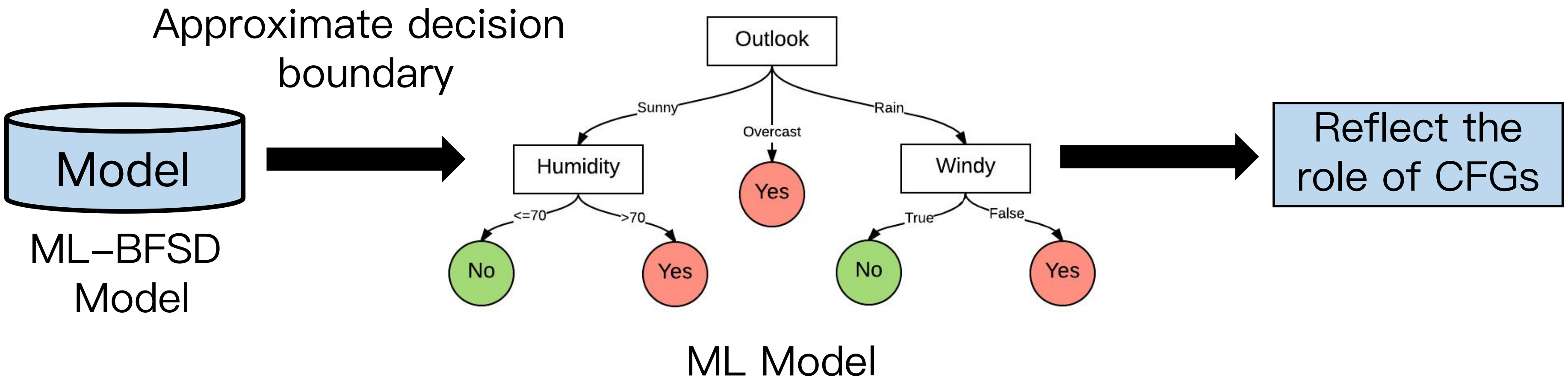
Design

➢ We design Explainer to explain BFSD and reveal the role of CFGs:



Design

➤ We design Explainer to explain BFSD and reveal the role of CFGs:

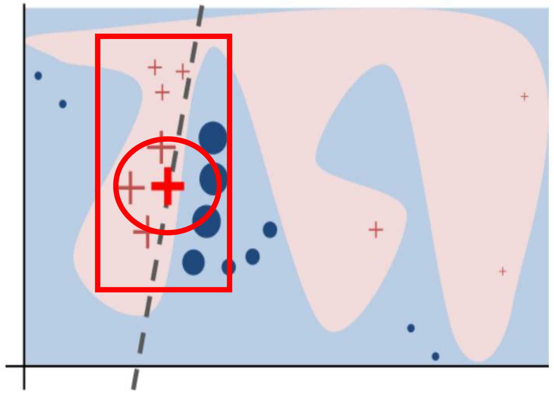


How to specify human-readable features?

Semantic Features	Call	Jump	Arithmetic
	Data Transfer	Other	
CFG Features	No. of Nodes	No. of Edges	Graph Similarity

Feature Specification

How to approximate decision boundary?



Local Approximation

Result

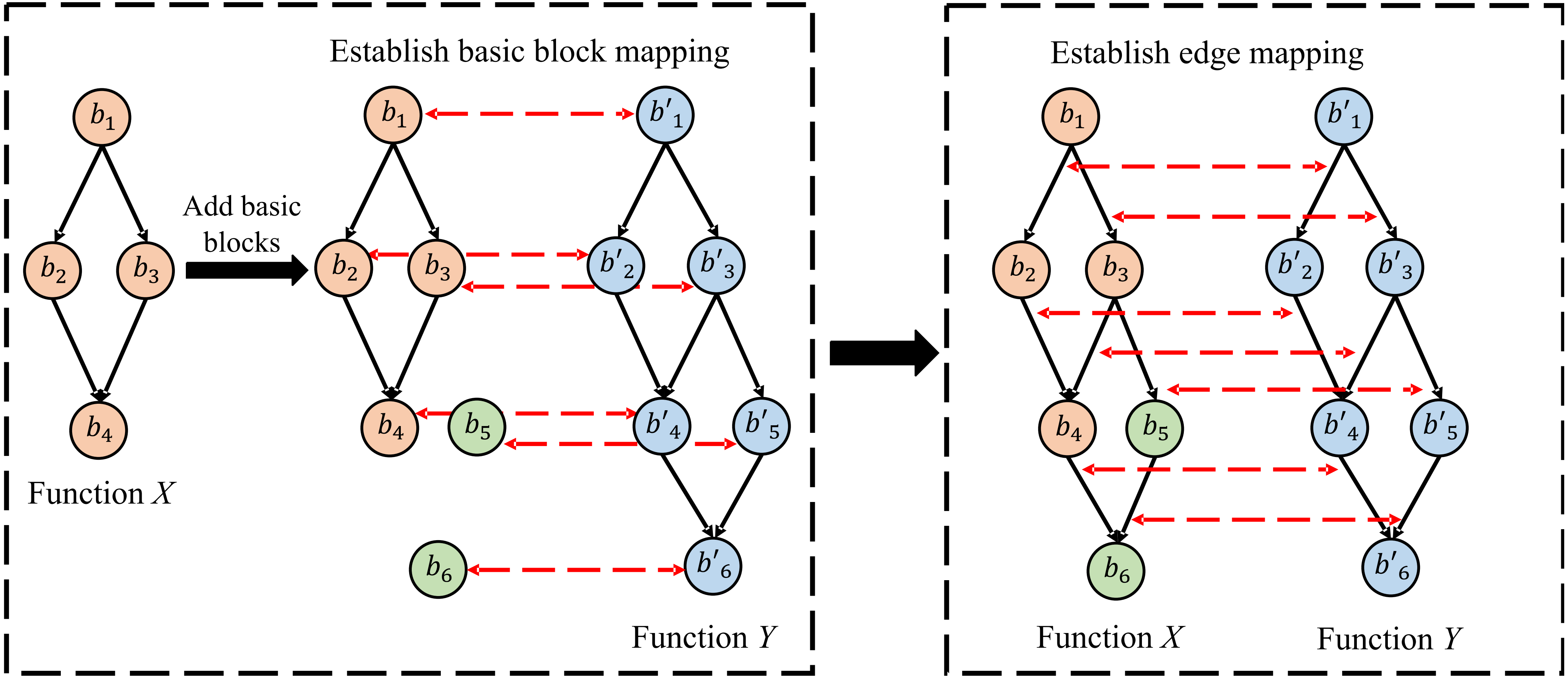
- Importance of different features:

Table 2: Evaluation results of average importance scores on each similarity detection solution.

Explanation Method	BFSD Solutions	Average Score								CFG features score the highest?
		Semantic Features					CFG Features			
		Call	Jump	Arith	Data-Tran	Other	Nodes	Edges	Graph-Sim	
LIME	Genius	0.071	0.098	0.032	0.051	0.073	0.127	0.119	**0.144**	Yes
	Asm2Vec	0.063	0.088	0.046	0.055	0.085	0.116	0.124	**0.216**	Yes
	Gemini	0.056	0.109	0.054	0.050	0.065	0.142	0.143	**0.381**	Yes
	GMN	0.058	0.062	0.074	0.067	0.062	0.156	0.138	**0.384**	Yes
	GraphEmb	0.079	0.107	0.073	0.072	0.113	0.155	0.121	**0.278**	Yes
	OrderMatters	0.095	0.074	0.110	0.083	0.093	0.171	0.141	**0.234**	Yes
	XBA	0.154	0.118	0.100	0.103	0.108	0.119	0.108	**0.190**	Yes
	DEXTER	0.117	0.126	0.109	0.108	0.116	0.152	0.119	**0.163**	Yes
	SAFE	0.102	0.119	0.149	0.115	**0.152**	0.129	0.095	0.140	No
	Trex	0.115	0.118	0.128	0.122	**0.136**	0.130	0.122	0.130	No
	jTrans	0.127	**0.171**	0.108	0.124	0.129	0.117	0.106	0.126	No
LEMNA	Genius	0.088	0.096	0.074	0.082	0.088	0.118	0.109	**0.179**	Yes
	Asm2Vec	0.085	0.104	0.085	0.112	0.088	0.121	0.114	**0.182**	Yes
	Gemini	0.080	0.146	0.079	0.070	0.097	0.111	0.125	**0.291**	Yes
	GMN	0.090	0.097	0.113	0.109	0.102	0.152	0.104	**0.233**	Yes
	GraphEmb	0.108	0.147	0.102	0.103	0.155	0.111	0.100	**0.174**	Yes
	OrderMatters	0.061	0.105	0.072	0.080	0.093	0.168	0.143	**0.223**	Yes
	XBA	0.147	0.131	0.110	0.113	0.121	0.108	0.124	**0.186**	Yes
	DEXTER	0.120	0.124	0.113	0.114	0.122	0.146	0.123	**0.152**	Yes
	SAFE	0.108	0.126	0.158	0.124	**0.169**	0.103	0.091	0.121	No
	Trex	0.118	0.121	0.131	0.126	**0.133**	0.127	0.121	0.123	No
	jTrans	0.132	**0.162**	0.112	0.127	0.136	0.110	0.103	0.118	No

Design

- We propose δCFG to assess the impact of CFGs by making them identical or different:



Basic block match algorithm **Edge match algorithm**

Result

Tsinghua University

- Given a function pair with **identical semantics**, we manipulate their CFGs to be **different**.
- Given a function pair with **different semantics**, we manipulate their CFGs to be **identical**.

BFSD Solutions	ER (%)			
	pool size = 16	pool size = 32	pool size = 64	pool size = 128
Genius	62.7	67.9	73.3	75.1
Asm2Vec	31.7	37.4	43.8	48.0
Gemini	40.1	49.2	57.4	65.3
GMN	42.2	47.6	54.7	64.1
GraphEmb	38.7	45.6	52.1	60.2
OrderMatters	57.8	65.1	70.7	75.2
XBA	52.0	62.5	70.6	76.0
DEXTER	46.5	56.3	63.7	70.6
SAFE	1.4	1.8	2.0	2.5
Trex	1.3	1.5	2.2	4.1
jTrans	1.1	1.5	2.4	3.0

When CFGs become different

BFSD Solutions	ER (%)			
	pool size = 16	pool size = 32	pool size = 64	pool size = 128
Genius	72.1	82.1	85.3	88.4
Asm2Vec	51.0	53.5	56.1	59.1
Gemini	52.5	58.6	63.1	71.2
GMN	50.8	61.6	67.1	71.9
GraphEmb	43.5	49.8	55.2	62.2
OrderMatters	69.2	74.3	78.1	81.3
XBA	59.8	69.1	76.5	79.6
DEXTER	58.9	64.3	68.6	74.0
SAFE	2.0	3.2	3.4	4.4
Trex	1.6	2.1	2.4	3.1
jTrans	1.5	1.7	2.6	3.1

When CFGs become identical

Result

Tsinghua University

- Given a function pair with **identical semantics**, we manipulate their CFGs to be **different**.
- Given a function pair with **different semantics**, we manipulate their CFGs to be **identical**.

BFSD Solutions	ER (%)			
	pool size = 16	pool size = 32	pool size = 64	pool size = 128
Genius	62.7	67.9	73.3	75.1
Asm2Vec	31.7	37.4	43.8	48.0
Gemini	40.1	49.2	57.4	65.3
GMN	42.2	47.6	54.7	64.1
GraphEmb	38.7	45.6	52.1	60.2
OrderMatters	57.8	65.1	70.7	75.2
XBA	52.0	62.5	70.6	76.0
DEXTER	46.5	56.3	63.7	70.6
SAFE	1.4	1.8	2.0	2.5
Trex	1.3	1.5	2.2	4.1
jTrans	1.1	1.5	2.4	3.0

When CFGs become different

BFSD Solutions	ER (%)			
	pool size = 16	pool size = 32	pool size = 64	pool size = 128
Genius	72.1	82.1	85.3	88.4
Asm2Vec	51.0	53.5	56.1	59.1
Gemini	52.5	58.6	63.1	71.2
GMN	50.8	61.6	67.1	71.9
GraphEmb	43.5	49.8	55.2	62.2
OrderMatters	69.2	74.3	78.1	81.3
XBA	59.8	69.1	76.5	79.6
DEXTER	58.9	64.3	68.6	74.0
SAFE	2.0	3.2	3.4	4.4
Trex	1.6	2.1	2.4	3.1
jTrans	1.5	1.7	2.6	3.1

When CFGs become identical

Why do these models rely on CFGs?

Explanation

Tsinghua University

- Interpreting CFG over-reliance:

Design flaws

(1) Some neglect the order of instructions.
(2) Some learn intra-block semantics but not inter-block relation.
(3) Some partially learns relationships.

……

Explanation

Tsinghua University

➢ Interpreting CFG over-reliance:

Design flaws

(1) Some neglect the order of instructions.
(2) Some learn intra-block semantics but not inter-block relation.
(3) Some partially learns relationships.

……

Bias of training set

The proportion of four types of function pairs.

Repetition Count	Proportion (%)			
	Type 1	Type 2	Type 3	Type 4
Repetition #1	49.71	40.49	9.51	0.29
Repetition #2	49.73	42.46	7.54	0.27
Repetition #3	49.75	39.39	10.61	0.25
Repetition #4	49.72	40.31	9.69	0.28
Repetition #5	49.68	39.33	10.67	0.32

(1) Type 1: different CFGs and different semantics.
(2) Type 2: different CFGs but same semantics.
(3) Type 3: same CFGs and same semantics.
(4) Type 4: same CFGs but different semantics.

Explanation

- Interpreting CFG over-reliance:

Design flaws

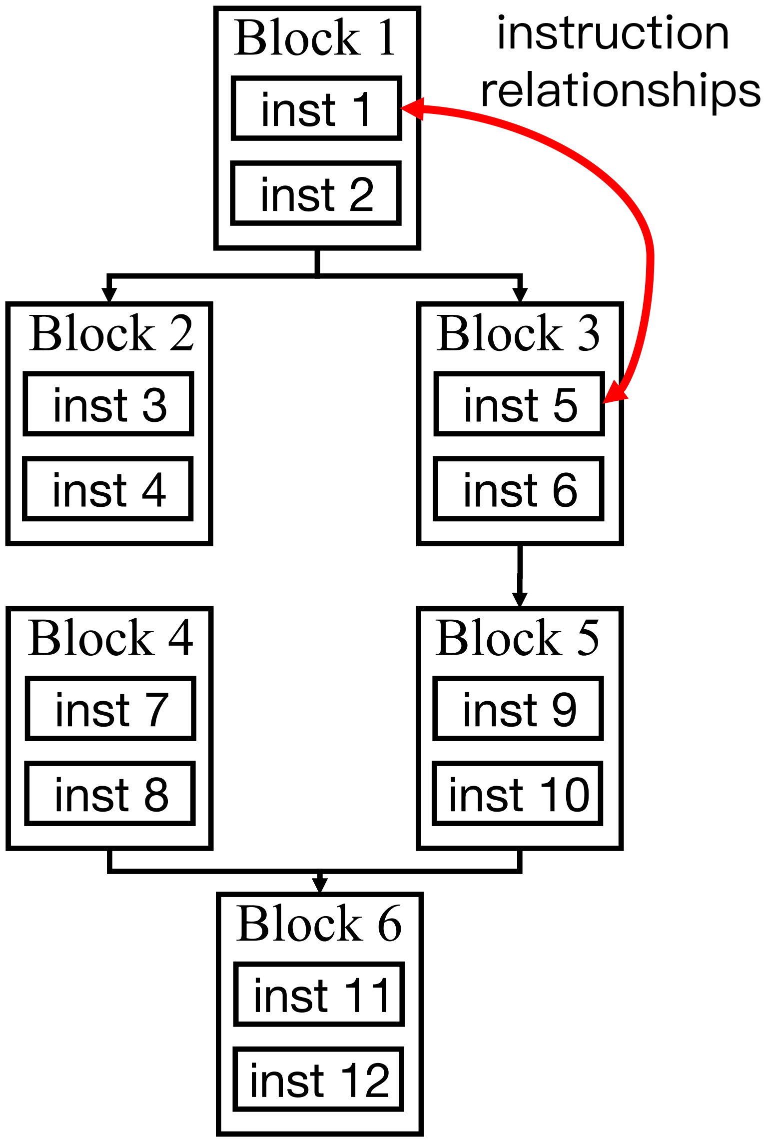
(1) Some neglect the order of instructions.
(2) Some learn intra-block semantics but not inter-block relation.
(3) Some partially learns relationships.
……

Bias of training set

The proportion of four types of function pairs.

Repetition Count	Proportion (%)			
	Type 1	Type 2	Type 3	Type 4
Repetition #1	49.71	40.49	9.51	0.29
Repetition #2	49.73	42.46	7.54	0.27
Repetition #3	49.75	39.39	10.61	0.25
Repetition #4	49.72	40.31	9.69	0.28
Repetition #5	49.68	39.33	10.67	0.32

(1) Type 1: different CFGs and different semantics.
(2) Type 2: different CFGs but same semantics.
(3) Type 3: same CFGs and same semantics.
(4) Type 4: same CFGs but different semantics



Explanation

- Interpreting CFG over-reliance:

Design flaws

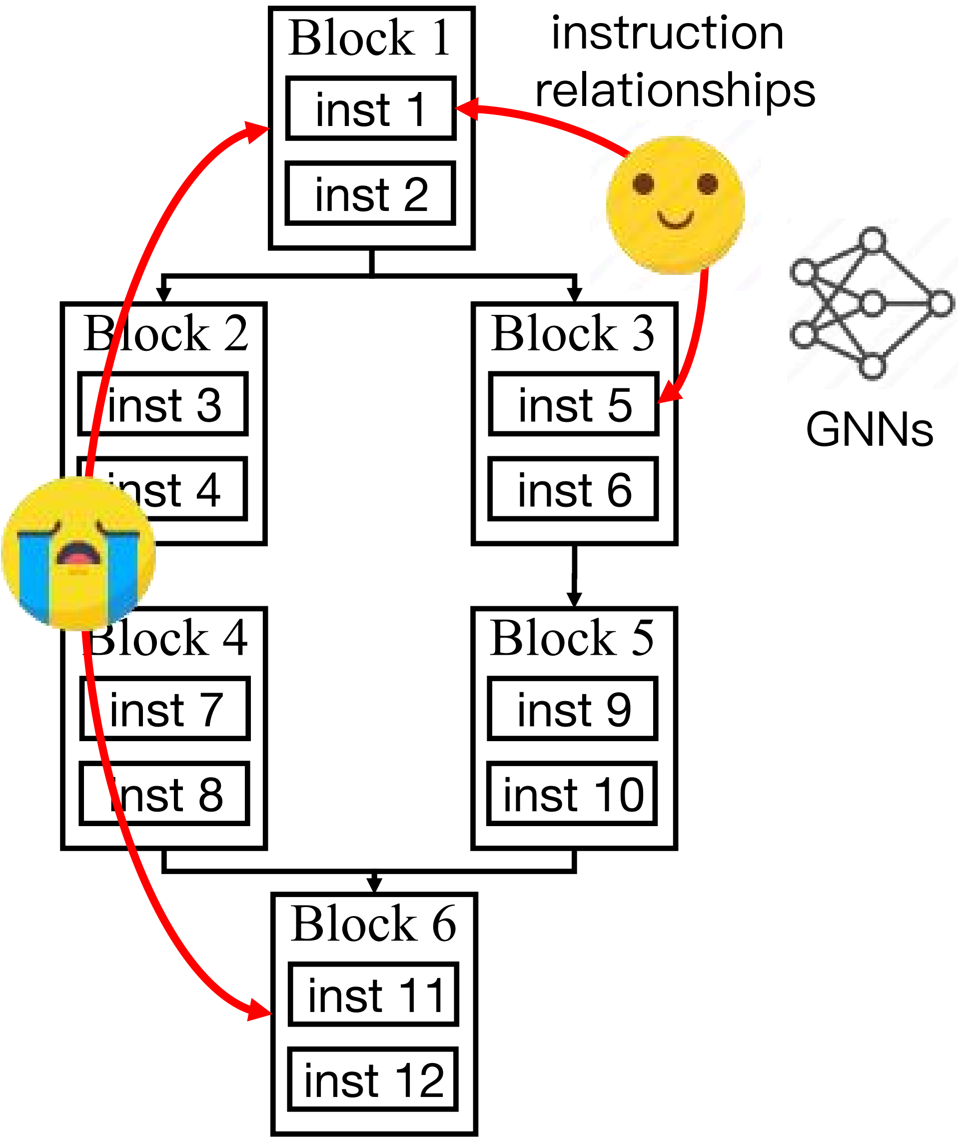
(1) Some neglect the order of instructions.
(2) Some learn intra-block semantics but not inter-block relation.
(3) Some partially learns relationships.
……

Bias of training set

The proportion of four types of function pairs.

Repetition Count	Proportion (%)			
	Type 1	Type 2	Type 3	Type 4
Repetition #1	49.71	40.49	9.51	0.29
Repetition #2	49.73	42.46	7.54	0.27
Repetition #3	49.75	39.39	10.61	0.25
Repetition #4	49.72	40.31	9.69	0.28
Repetition #5	49.68	39.33	10.67	0.32

(1) Type 1: different CFGs and different semantics.
(2) Type 2: different CFGs but same semantics.
(3) Type 3: same CFGs and same semantics.
(4) Type 4: same CFGs but different semantics



Result

- We improve models' preformance in BFSD by finetuning with δCFG:

Table 7: Comparison of improvement in MRR after fine-tuning with and without augmented data (denoted as **clean**), expressed as Δ MRR. The results validate the effectiveness of using δCFG for fine-tuning the models.

BFSD Solutions	**Δ MRR (%)**													
	O0,O3		**O1,O3**		**O2,O3**		**O0,Os**		**O1,Os**		**O2,Os**		**Average**	
	δCFG	**clean**	**δCFG**	**clean**	**δCFG**	**clean**	**δCFG**	**clean**	**δCFG**	**clean**	**δCFG**	**clean**	**δCFG**	**clean**
Gemini	**9.0**	0.0	**3.5**	0.2	**1.6**	0.2	**7.0**	0.2	**5.5**	-0.1	**4.8**	0.0	**5.2**	0.1
GMN	**10.1**	0.3	**5.0**	0.2	**1.4**	0.2	**9.8**	0.1	**4.0**	-0.1	**4.1**	0.3	**5.7**	0.2
GraphEmb	**3.7**	-0.1	**2.8**	0.2	**1.0**	0.3	**3.6**	-0.1	**2.9**	0.1	**3.1**	0.1	**2.9**	0.1
OrderMatters	**1.4**	0.1	**1.3**	-0.1	**0.9**	0.1	**3.1**	0.1	**1.3**	-0.1	**1.8**	0.1	**1.6**	0.1
XBA	**0.4**	0.1	**0.3**	-0.1	**0.2**	0.1	**1.1**	0.1	**0.8**	0.1	**0.4**	0.1	**0.5**	0.1
DEXTER	**1.4**	-0.2	**4.7**	-0.1	**1.9**	0.2	**2.0**	0.3	**4.5**	0.1	**1.4**	0.3	**2.7**	0.1

Table 8: Comparison of improvement in Recall@1 after fine-tuning with and without augmented data (denoted as **clean**), expressed as Δ Recall@1. The results validate the effectiveness of δCFG.

BFSD Solutions	**Δ Recall@1 (%)**													
	O0,O3		**O1,O3**		**O2,O3**		**O0,Os**		**O1,Os**		**O2,Os**		**Average**	
	δCFG	**clean**	**δCFG**	**clean**	**δCFG**	**clean**	**δCFG**	**clean**	**δCFG**	**clean**	**δCFG**	**clean**	**δCFG**	**clean**
Gemini	**10.8**	-0.1	**4.4**	0.2	**1.9**	0.1	**9.3**	0.0	**6.7**	0.1	**5.6**	0.1	**6.5**	0.1
GMN	**12.7**	0.3	**7.1**	-0.1	**1.8**	0.2	**12.7**	0.1	**6.2**	0.3	**5.9**	0.1	**7.7**	0.2
GraphEmb	**5.2**	0.1	**4.3**	0.2	**1.5**	0.3	**5.4**	0.2	**4.2**	0.3	**4.3**	0.2	**4.2**	0.3
OrderMatters	**1.8**	0.0	**1.6**	-0.1	**1.2**	0.1	**3.5**	0.2	**1.7**	0.1	**2.5**	0.0	**2.1**	0.1
XBA	**0.6**	0.0	**0.4**	-0.1	**0.3**	0.1	**1.6**	0.1	**1.3**	0.2	**0.7**	0.0	**0.8**	0.1
DEXTER	**1.1**	-0.3	**7.0**	0.2	**3.4**	-0.2	**3.3**	0.3	**7.1**	0.3	**1.8**	0.3	**3.9**	0.1

Result

➢ We lower the ER by finetuning with δCFG:

BFSD Solutions	ER (%)							
	pool size = 16		pool size = 32		pool size = 64		pool size = 128	
	δCFG	baseline	δCFG	baseline	δCFG	baseline	δCFG	baseline
Gemini	**30.6**	36.0	**36.8**	42.8	**40.9**	47.4	**46.1**	51.6
GMN	**23.0**	33.5	**26.4**	37.3	**29.2**	41.7	**32.8**	46.1
GraphEmb	**23.2**	47.1	**28.3**	51.4	**33.7**	54.6	**38.7**	57.3
OrderMatters	**13.5**	29.6	**18.6**	35.5	**24.2**	43.3	**28.8**	50.5
XBA	**47.6**	51.5	**53.9**	57.4	**58.8**	61.5	**62.0**	65.2
DEXTER	**40.4**	49.3	**43.2**	55.5	**47.9**	60.1	**53.1**	62.8

清華大學 Tsinghua University

Improving ML-based Binary Function Similarity Detection by Assessing and Deprioritizing Control Flow Graph Features

Jialai Wang*, Chao Zhang*‡§ , Longfei Chen*, Yi Rong*, Yuxiao Wu†, Hao Wang*, Wende Tan*, Qi Li*, and Zongpeng Li*♣§
*Tsinghua University, †Huazhong University of Science and Technology,
‡Zhongguancun Laboratory, ♣Quancheng Labs
{wang-jl22, clf23, rongy19, hao-wang20}@mails.tsinghua.edu.cn, wyxhustcse@hust.edu.cn,
twd2.me@gmail.com, {chaoz, qli01, zongpeng}@tsinghua.edu.cn

➢ **Key Contributions:**
- Assess the role of CFG features in ML-BFSD models
- Apply explanation methods to ML-BFSD models and reveal heavy reliance on CFG features in existing models
- Propose CFG manipulation solution (δCFG)
- Improve performance of existing models