



Fuzzlang: Transformer and LLM-Agent for Enhanced Compilation Error Repair

Baodi Shan¹, Barbara Chapman¹, Johannes Doerfert² Mar 1, 2025

- 1 Stony Brook University
- 2 Lawrence Livermore National Laboratory

Problem Statement

Compilation Error





How to fix compilation errors?

YOU CANT HAVE COMPLICATION ERRORS

IF YOU NEVER COMPILE



Related Work

Code Fix for C/C++ Compilation Error -- Traditional Methods

- Compiler itself
 - Detailed Diagnostics
 - Enhanced Error Messages
 - Fix-it Hints
- IDE Real Time Feedback

- Static Analysis
- Cross-file Error Contexts
- Incremental Compilation

Code Fix for C/C++ Compilation Error -- Traditional Methods

- Compiler itself
 - **Detailed Diagnostics**
 - **Enhanced Error Messages**
 - **Fix-it Hints**
- IDF Real Time Feedback



Google

- Static Analysis
- Cross-file Frror Contexts
- **Incremental Compilation**



Baodi Shan 9:00 AM

Shilei, could you help me?



Code Fix for C/C++ Compilation Error -- ML Method

- → Deepfix (2017 AAAI)
 - ◆ GRU (Gated Recurrent Unit)
- → TRACER(2018 ICSE-SEET)
 - ◆ RNN
- → DrRepair (2020 ICML), Break-It-Fix-It (2021 ICML)
 - ◆ LSTM
- →

Table 2. A summary and comparison of representative learning-based APR approaches

Year	Technique	Type	Language	Localization	Abstraction	Context	Tokenization	Representation	Model	Ranking
2016	Bhatia et al. [18]	Syntax	Python	Perfect	No	Method	word	token	RNN	N.A.
1017	Deepfix [58]	Syntax	C	SD	No	Method	N.A.	token	GRU	N.A.
017	Wang et al. [191]	Semantic	C	N.A.	No	Method	N.A.	token	RNN	N.A.
2017	VuRLE [116]	Vulnerability	Java	SD	Yes	Statement.	N.A.	graph	N.A.	N.A.
8100	Harer et al. [62]	Vulnerability	C.C++	N.A.	No	Method	N.A.	token	GAN	N.A.
8105	TRACER [6]	Syntax	C	SD	Yes	Method	N.A.	token	RNN	beam search
8100	Santos et al. [167]	Syntax	Java	SD	Yes	Method	N.A.	token	LSTM	patch re-ranking
018	Bhatia et al. [17]	Syntax	Python	N.A.	No	Method	N.A.	token	RNN	patch re-ranking
018	Sarfgen [192]	Syntax	C	N.A.	No	Method	N.A.	tree	N.A.	patch filtering & re-ranking
019	SequenceR [27]	Semantic	Java	Perfect	Yes	Class	word	token	LSTM	beam search
019	Codit [23]	Syntax	Java	Perfect	Yes	Method	N.A.	tree	Tree-LSTM	beam search
019	Tufano et al. [183]	Semantic	Java	N.A.	Yes	Method	word	token	RNN	beam search
1019	Tufano et al. [184]	Semantic	Java	Perfect	perfect	Method	word	token	RNN	RNN
2019	Chen et al. [27]	Semantic	Java	N.A.	No	Class	N.A.	token	RNN	N.A.
019	DeepDelta [132]	Syntax	Java	Perfect	Yes	Method	N.A.	tree	LSTM	beam search
019	RLAssitst [57]	Syntax	C	SD	No	Method	N.A.	token	LSTM	N.A.
020	CoCoNut [115]	Semantic	Java,C.Python,JS	Perfect	Yes	Method	word	token	FConv	beam search
1020	DLFix [98]	Semantic	Java	SBFL	Yes	Method	word	tree	Tree-LSTM	patch filtering & re-ranking
020	DrRepair [222]	Syntax	C.C++	SD	No	Method	N.A.	graph	LSTM	N.A.
1020	Hoppity [39]	Semantic	35	SD	No	Statement	N.A.	graph	LSTM	beam search
1020	Yang et al. [219]	Syntax	c	SD	N.A.	Method	subword	token	SegGAN	patch re-ranking
1020	GGF [205]	Syntax	C	SD	No	Method	N.A.	token,graph	GGNN	N.A.
1021	CURE [73]	Semantic	Java	Perfect	No	Method	subword	token	GPT	code-aware beam search
021	Recoder [242]	Syntax	Java	SBFL Perfect	No	Method	word	graph	Tree-LSTM	beam search
021	TFix [15]	Semantic	IS	Perfect	No	Statement	subword	token	TS	beam search
1021	GrasP [175]	Semantic	Java	Perfect	No	Method	word	graph	RNN,GNN	beam search
021	SampleFix [60]	Syntax	C	SD	No	Method	N.A.	token	LSTM	beam search
022	CIRCLE [228]	Semantic	Java,C.JS,Python	Perfect	No	Method	subword	token	TS	beam search
1022	DEAR [99]	Semantic	Java	SBFL	Yes	Statement	word	tree	Tree-LSTM	patch filtering & re-ranking
1022	Graphix [142]	Semantic	Java	Perfect	Yes	Method	N.A.	graph,tree	Tree-LSTM	N.A.
1022	SelfAPR [226]	Semantic	Java	Perfect	No	Method	subword	token	Transformer	beam search
022	VRepair [28]	Vulnerability	C	Perfect	No	Method	word	token	Transformer	beam search
1022	SegTrans [31]	Vulnerability	Java	Perfect	Yes	Statement	subword	token	Transformer	beam search
022	AlphaRepair [209]	Semantic	Java Python	Perfect	No	Class	subword	token	CodeBERT	CodeBERT re-ranking
022	VulRepair [51]	Vulnerability	C	Perfect	No	Method	subword	token	T5	beam search
022	Bug-Transformer [221]	Semantic	Java	Perfect	Yes	Method	subword	token	Transformer	beam search
022	SPVF [241]	Vulnerability	C++,C,Python	Perfect	No	Method	N.A.	tree	Transformer	beam search patch filtering
1022	SYNSHINE [4]	Syntax	Java	SD	Yes	Class	subword	token	Transformer	N.A.
1022	MMAPR [231]	Semantic,Syntax	Python	Perfect	No	Class	subword	token	Codex	N.A.
1022	RING [75]	Syntax	Python JS,C	SD	No	Method	subword	token	Codex	patch re-ranking
1022	RewardRepair [227]	Semantic	Java	SBFL Perfect	No	Statement	subword	token	Transformer	beam search
022	BIFI [223]	Syntax	Python,C	N.A.	No	Method	N.A.	token graph	LSTM	beam search

Quanjun Zhang, Chunrong Fang, Yuxiang Ma, Weisong Sun, and Zhenyu Chen. 2023. A Survey of Learning-based Automated Program Repair. ACM Trans. Softw. Eng.

Methodol. 33, 2, Article 55 (February 2024), 69 pages. https://doi.org/10.1145/3631974

ar .	Technique	Type	Language	Localization	Abstraction	Context	Tokenization	Representation	Model	Ranking	urvey of Learning
6	Bhatia et al. [18]	Syntax	Python	Perfect	No	Method	word	token	RNN	N.A.	2
7	Deepfix [58]	Syntax	C	SD	No	Method	N.A.	token	GRU	N.A.	3.
	Wang et al. [191]	Semantic	C	N.A.	No	Method	N.A.	token	RNN	N.A.	8
7	VuRLE [116]	Vulnerability	Java	SD	Yes	Statement	N.A.	graph	N.A.	N.A.	ġr.
	Harer et al. [62]	Vuln									
8	TRACER [6]										
	Santos et al. [167]	/									
8	Bhatia et al. [17] Sarfgen [192]	1									
9	SequenceR [27]										
9	Codit [23]										
	Tufano et al. [183]										
	Tufano et al. [184]										
9	Chen et al. [27]										
	DeepDelta [132]										
9	RLAssitst [57]										
10	CoCoNut [115]				_				_		
			43 15 4								
50	DLFix [98]										
10	DLFix [98] DrRepair [222]									/WXC)	
	DLFix [98] DrRepair [222] Hoppity [39]		are	45		u	IGI	JUC	15	MO	uei
10	DrRepair [222]		arg	ye		u	191	Jui	15	Mo	uei
10	DrRepair [222] Hoppity [39]	L	ar	ye	L	u	191	JU	JE	MO	uei
10 10 10 10	DrRepair [222] Hoppity [39] Yang et al. [219] GGF [205] CURE [73]	L	arg	ye	L	u	191	Jui	je	MO	uei
10 10 10 10 11	DrRepair [222] Hoppity [39] Yang et al. [219] GGF [205] CURE [73] Recoder [242]		ar	ye	L	ur	igi	JU	je	1410	uei
10 10 10 10 11 11	DrRepair [222] Hoppity [39] Yang et al. [219] GGF [205] CURE [73] Recoder [242] TFix [15]		ar	ge	L	ur	191	Jug	je	1410	uei
10 10 10 10 11 11	DrRepair [222] Hoppity [39] Yang et al. [219] GGF [205] CURE [73] Recoder [242] TFix [15] GrasP [175]	L	arg	ge	L	ur	191	udi	je	MO	uei
10 10 10 10 11 11 11	DrRepair [222] Hoppity [39] Yang et al. [219] GGF [205] CURE [73] Recoder [242] TFix [15] GrasP [175] SampleFix [60]		arg	ge	L	ur	igi	uuç	je	74(0)	uei
10 10 10 10 11 11 11 11 11	DrRepair [222] Hoppty [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] GrasP [175] SampleFix [60] CIRCLE [228]	L	arg	ge	L	ur	igi	uuç	je	14(0)	uei
10 10 10 10 11 11 11 11 11 12 12	DrRepair [222] Hoppity [39] Yang et al. [219] GGF [205] CURE [73] Recoder [242] TFix [15] GrasP [175] SampleFix [60] CIRCLE [228] DEAR [59]	L	arg	ge	L	ur	igi	uug	je	14101	uei
00 00 00 00 01 11 11 11 11 11 12 22 22 22	DrRepair [222] Hoppity [39] Yang et al [219] GGF [205] CURE [72] Recoder [242] TFix [15] Grash [775] SampleFix [60] CIRCLE [228] DEAR [90] Graphix [142]	L	arg	ge	L	ar	igi	uug	je	14(0)	uei
00 00 00 01 11 11 11 11 12 12 22 22 22	DrRepair [222] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [342] TFix [15] Grash [175] Grash [175] DEAR [99] OTRICLE [228] DEAR [99] SelfAPR [226]		arg	ge	L	ur	igi	Jug	je	1410	uei
00 00 00 01 11 11 11 11 12 12 12 12 12 12 12 12 12	DrRepair [222] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] GrasP [175] SampleFix [60] CIRCLE [228] DEAR [79] Graphix [142] SelfAPR [226] VRepair [28]		arg	ge	L	ur	igi	Jug	je	1410	uei
00 00 00 00 11 11 11 11 11 11 12 22 22 22 22 22	DrRepair [222] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] Grash [175] SampleFix [60] CIRCLE [228] DEAR [89] Graphix [142] SelfAPR [226] VRepair [28] SegTrans [31]		arg	ge	ا	ar	igi	Ju	je	74(0)	uei
00 00 00 00 00 00 00 00 00 00 00 00 00	DrRepair [222] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] Gearl [175] SampleFix [60] CIRCLE [228] DEAR [99] Graphix [142] SelfAPR [226] VRepair [28] SeqTrans [31] UphaRepair [209]	Vi	Javes 1 (100)	1000	240	CHO			CORUGN	evocere re nasong	uei
000 000 000 000 000 000 000 000 000 00	DrRepair [222] Hoppity [39] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] GrasP [175] SampleFix [60] CIRCLE [228] DEAR [60] Graphix [142] SelfAPR [226] VRepair [28] SeqTrans [31] UphaRepair [209]	Va Semanus Vulnerability	And Janes	Perfect	No No	Method	subweed	boken	TS	beam search	uei
00 00 00 00 00 00 00 00 00 00 00 00 00	DrRepair [222] Hoppity [39] Yang et al [219] GGF [205] CURE [72] Recoder [242] TFix [15] Grash [175] SampleFix [60] CIRCLE [228] DEAR [90] Graphix [142] SelfAPR [226] VRepair [28] SeqTrans [31] MphaRepair [200] VulRepair [21] g-Transformer [221]	Vo. Semantic Vulnerability Semantic	C. Java	Perfect	No Yes	Method Method	subword	token token	TS Transformer	beam search beam search	uei
00 00 00 00 00 00 00 00 00 00 00 00 00	DrRepair [222] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] GeasP [175] SampleFix [60] CIRCLE [223] DEAR [99] Graphix [142] SelfAPR [226] VRepair [28] SelfAPR [276] VulRepair [28] VulRepair [51] SPVF [241]	Vilvenshiry Vulnerability Semantic Vulnerability	C Java C++,C.Python	Perfect Perfect Perfect	No Yes No	Method Method Method	subwerd subword N.A.	token token tree	Ts Transformer Transformer	beam search beam search beam search patch filtering	uei
200 000 000 000 000 000 000 000 000 000	DrRepair [222] Hoppity [39] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] Gash [175] SampleFix [60] CIRCLE [228] DEAR [99] Graphix [142] SelfAPR [226] VRepair [28] SeqTrans [31] UphaRepair [209] VulRepair [51] SPVF [241] SPVF [241]	Visconimo Vulnerability Semantic Vulnerability Syntax	C Java C++,C,Python Java	Perfect Perfect Perfect SD	No Yes No Yes	Method Method	subword	token token tree token	T5 Transformer Transformer Transformer	beam search beam search	uei
200 000 000 000 000 000 000 000 000 000	DrRepair [222] Hoppity [39] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] Grash [175] SampleFix [60] CIRCLE [228] DEAR [89] Graphix [142] SelfAPR [226] VRepair [28] SelfAPR [271ras [31] UphaRepair [209] VulRepair [21] STNSHINE [4] STNSHINE [4] MMAPR [251]	Volumerability Semantic Vulnerability Semantic Vulnerability Syntax	C Java C++,C,Python Java Python	Perfect Perfect Perfect	No Yes No Yes No	Method Method Method Class	subweed subword N.A. subword	token token tree	Ts Transformer Transformer	beam search beam search beam search, patch filtering N.A. N.A.	uei
000 000 000 000 000 000 000 000 000 00	DrRepair [222] Hoppity [39] Hoppity [39] Yang et al [219] GGF [205] CURE [73] Recoder [242] TFix [15] Gash [175] SampleFix [60] CIRCLE [228] DEAR [99] Graphix [142] SelfAPR [226] VRepair [28] SeqTrans [31] UphaRepair [209] VulRepair [51] SPVF [241] SPVF [241]	Visconimo Vulnerability Semantic Vulnerability Syntax	C Java C++,C,Python Java	Perfect Perfect Perfect SD Perfect	No Yes No Yes	Method Method Class Class	subweed subword N.A. subword subword	token token tree token token	Transformer Transformer Transformer Codex	beam search beam search beam search, patch filtering N.A.	uei

Table 2. A summary and comparison of representative learning-based APR approaches

Quanjun Zhang, Chunrong Fang, Yuxiang Ma, Weisong Sun, and Zhenyu Chen. 2023. A Survey of Learning-based Automated Program Repair. ACM Trans. Softw. Eng. 10 Methodol. 33, 2, Article 55 (February 2024), 69 pages. https://doi.org/10.1145/3631974

For LLMs, we need data

- → Deepfix -- Homework
- → C-Pack-IPAs -- Homework
- → CodeForces Dataset -- Online Judge(OJ)

Stack Overflow? Github? Other Public Resources?

In clang, how many compilation error types?

3541 unique error types! (commit 6441df3b)

C++/OpenMP/OpenACC/OpenCL are developing, more error types are coming!



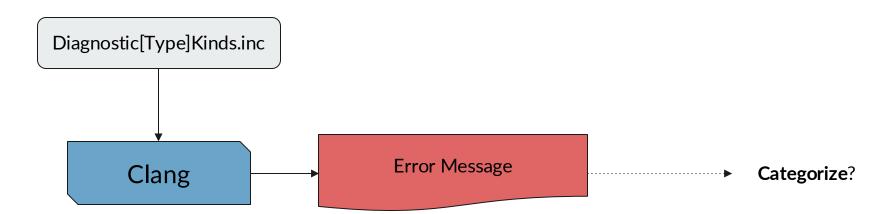
So, aims of Fuzzlang

- Let LLMs understand compilation error
- Categorize compilation error types
- Generate more compilation error

- ☐ Reproduce compilation error
- Cover more error types
- ☐ Fine-tuning current LLM

Implementation - Foundational Components

How clang diagnostic engine works?



Error Message?

```
expected %0 - "err_expected"

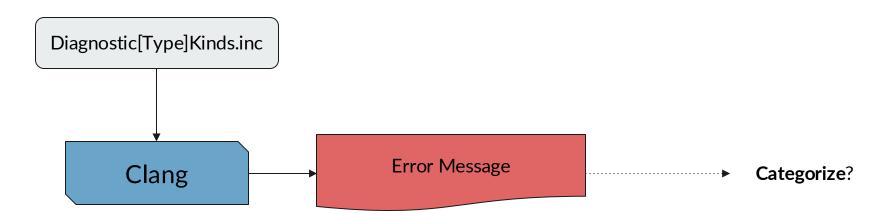
expected %0 or %1 - "err_expected_either"

expected %1 after %0 - "err_expected_after"
```

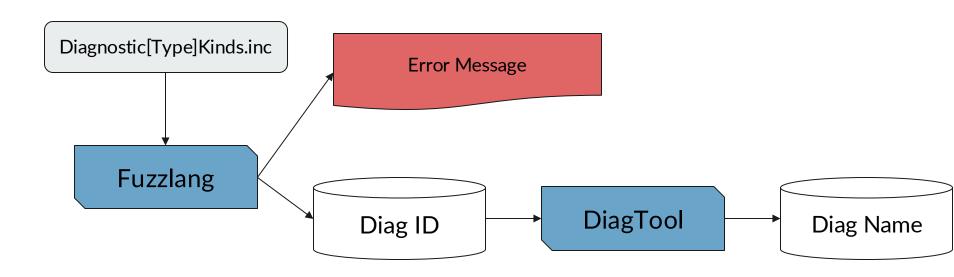
Fuzzlang Foundational Components

- Specially Modified Clang -- Diag ID
- Specially Modified Diagtool -- Diag Name

How clang diagnostic engine works?

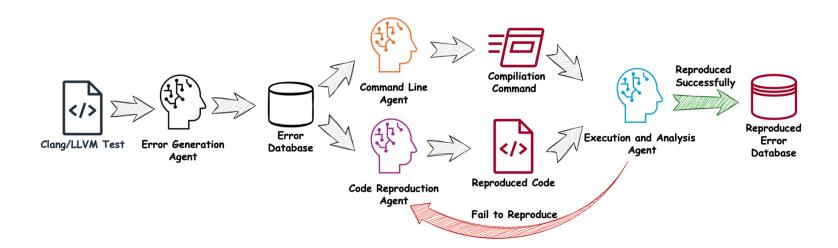


How Fuzzlang diagnostic engine works?



Implementation - Fuzzlang Agent

Workflow of Fuzzlang Agent



Implementation - Fuzzlang Transformer

Introduction of Fuzzing

What's Fuzzing?

Automated software testing technique that inputs invalid, unexpected, or random data into a program to find vulnerabilities and errors.

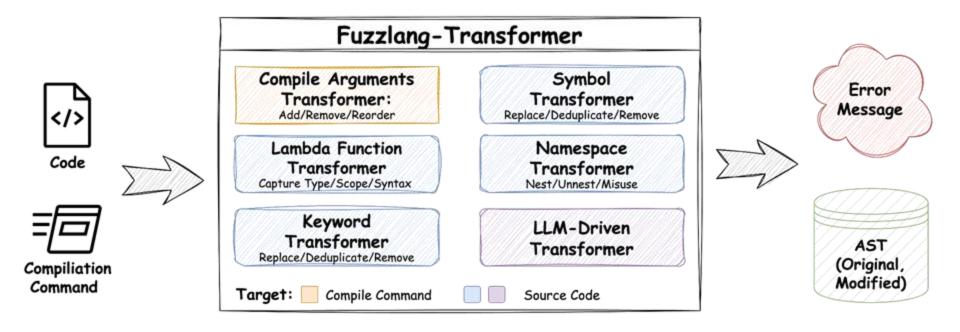
Workflow of Fuzzing?



Design of Fuzzlang Transformer

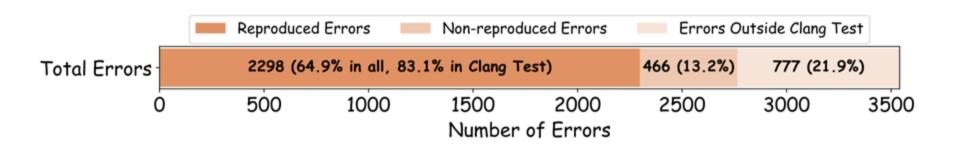
- A Clang Wrapper (Used as a C/C++ compiler)
- "Modification" Modules
 - Compilation Command
 - Source Code
- Collecting Error Message & AST

Workflow of Fuzzlang Transformer

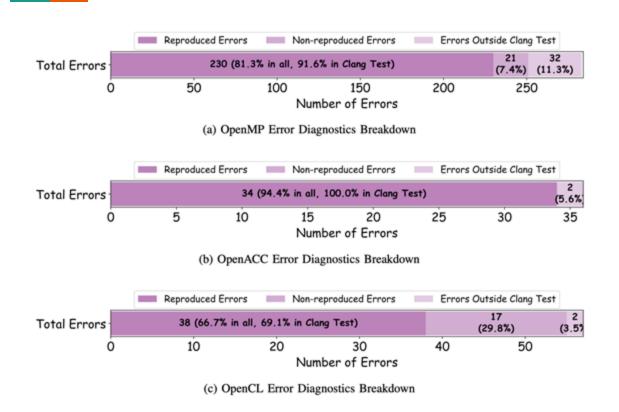


Evaluation

Error Diagnostics Reproducing Breakdown



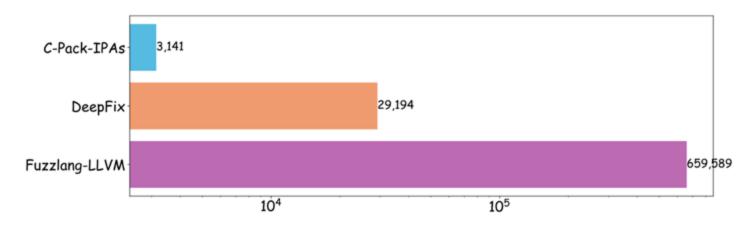
Bonus: Fuzzlang for HPC



Dataset: Fuzzlang-LLVM

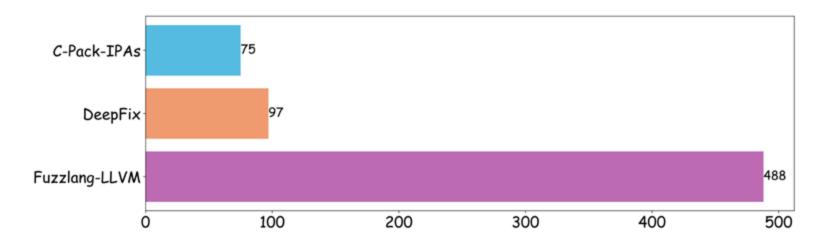
- Compile LLVM project(LLVM+Clang)
- ~4000 Compilation Steps
- 659k errors
- 488 Unique Types

Dataset: Fuzzlang-LLVM



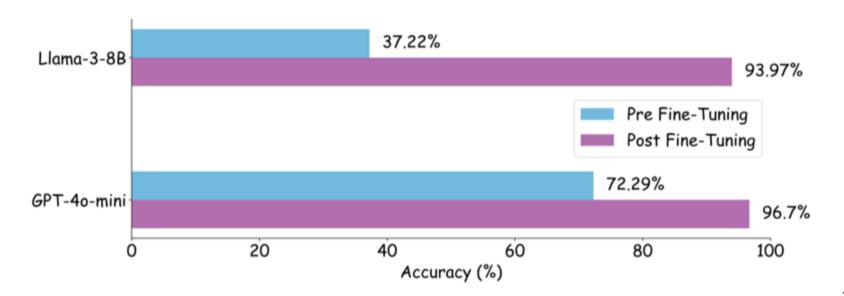
(a) Total Number of Errors (log)

Dataset: Fuzzlang-LLVM



(b) Number of Independent Error Types

Validation -- Fine Tuning LLM



Next Steps

What next?

- → Larger dataset from other C/C++ project
- → Modification cross files

Runtime Error Monitoring(Vulnerability)

Q&A

Contact: baodi.shan@stonybrook.edu