The 16th International Conference on Internetware (INTERNETWAE 2025)

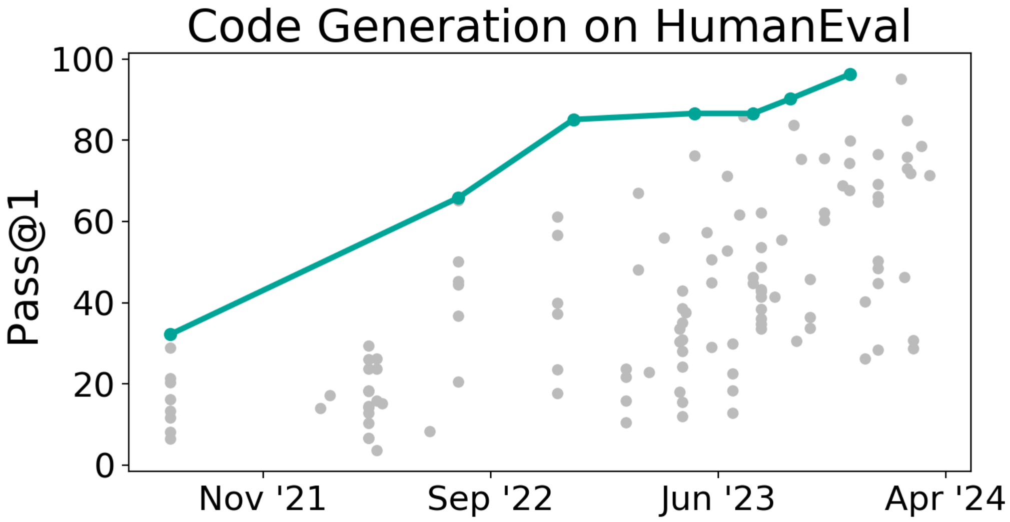
MetaCoder: Generating Code from Multiple Perspectives

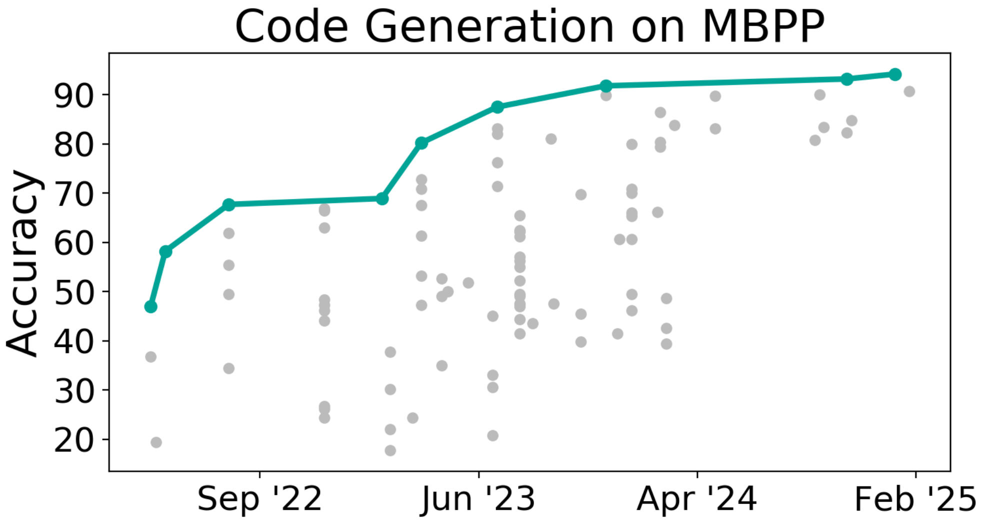
Xin Chen, Zhijie Jiang, Shanshan Li, Yong Guo, Zhouyang Jia, Si Zheng, Yuanliang Zhang

College of Computer Science and Technology
National University of Defense Technology
Changsha, China

Background

- **The code generation capability of LLMs has seen rapid advancements over the past few years**





Motivation

- **LLMs show the performance gap in generating different programming languages**

High-proficiency ← Gap → Low-proficiency

Model	Dataset	Python	C++	Java
Llama 3.1 8B Instruct	HumanEval	72.6 ±6.8	52.8 ±7.7	58.2 ±7.7
	MBPP	60.8 ±4.3	53.7 ±4.9	54.4 ±5.0
Llama 3.1 70B Instruct	HumanEval	80.5 ±6.1	71.4 ±7.0	72.2 ±7.0
	MBPP	75.4 ±3.8	65.2 ±4.7	65.3 ±4.8
Llama 3.1 405B Instruct	HumanEval	89.0 ±4.8	82.0 ±5.9	80.4 ±6.2
	MBPP	78.8 ±3.6	67.5 ±4.6	65.8 ±4.7
DS-Coder-V2-Lite-Instruct	HumanEval	81.1	75.8	76.6
DS-Coder-V2-Instruct	HumanEval	90.2	84.8	82.3

Performance metrics (%) for various LLMs on code generation benchmarks[1][2]

How to bridge the performance gap?

[1] Abhimanyu Dubey, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024).
[2] Qihao Zhu, et al. 2024. Deepseek-coder-v2:Breaking the barrier of closed-source models in code intelligence. arXiv preprint arXiv:2406.11931 (2024).

Motivation

- **LLMs show the performance gap in generating different programming languages**

High-proficiency ← Gap → Low-proficiency

Model	Dataset	Python	C++	Java
Llama 3.1 8B Instruct	HumanEval	72.6 ±6.8	52.8 ±7.7	58.2 ±7.7
	MBPP	60.8 ±4.3	53.7 ±4.9	54.4 ±5.0
Llama 3.1 70B Instruct	HumanEval	80.5 ±6.1	71.4 ±7.0	72.2 ±7.0
	MBPP	75.4 ±3.8	65.2 ±4.7	65.3 ±4.8
Llama 3.1 405B Instruct	HumanEval	89.0 ±4.8	82.0 ±5.9	80.4 ±6.2
	MBPP	78.8 ±3.6	67.5 ±4.6	65.8 ±4.7
DS-Coder-V2-Lite-Instruct	HumanEval	81.1	75.8	76.6
DS-Coder-V2-Instruct	HumanEval	90.2	84.8	82.3

Performance metrics (%) for various LLMs on code generation benchmarks[1][2]

How to bridge the performance gap?

Fine-tuning costs too much!

[1] Abhimanyu Dubey, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024).
[2] Qihao Zhu, et al. 2024. Deepseek-coder-v2:Breaking the barrier of closed-source models in code intelligence. arXiv preprint arXiv:2406.11931 (2024).

Motivation

- **LLMs show the performance gap in generating different programming languages**

High-proficiency ⟷ Gap ⟷ Low-proficiency

Model	Dataset	Python	C++	Java
Llama 3.1 8B Instruct	HumanEval	72.6 ±6.8	52.8 ±7.7	58.2 ±7.7
	MBPP	60.8 ±4.3	53.7 ±4.9	54.4 ±5.0
Llama 3.1 70B Instruct	HumanEval	80.5 ±6.1	71.4 ±7.0	72.2 ±7.0
	MBPP	75.4 ±3.8	65.2 ±4.7	65.3 ±4.8
Llama 3.1 405B Instruct	HumanEval	89.0 ±4.8	82.0 ±5.9	80.4 ±6.2
	MBPP	78.8 ±3.6	67.5 ±4.6	65.8 ±4.7
DS-Coder-V2-Lite-Instruct	HumanEval	81.1	75.8	76.6
DS-Coder-V2-Instruct	HumanEval	90.2	84.8	82.3

Performance metrics (%) for various LLMs on code generation benchmarks[1][2]

How to bridge the performance gap?

Fine-tuning costs too much!

Why not LLM guides itself?

- High-proficiency language guides low-proficiency language

[1] Abhimanyu Dubey, et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024).
[2] Qihao Zhu, et al. 2024. Deepseek-coder-v2:Breaking the barrier of closed-source models in code intelligence. arXiv preprint arXiv:2406.11931 (2024).

A Motivating Example

- **Using high-proficiency language (Python) to guide the generation of low-proficiency language (C++)**

Task Description

You are given a 2D integer array points and an integer w.Your task is to cover all the given points with rectangles. Return the minimum number of rectangles.

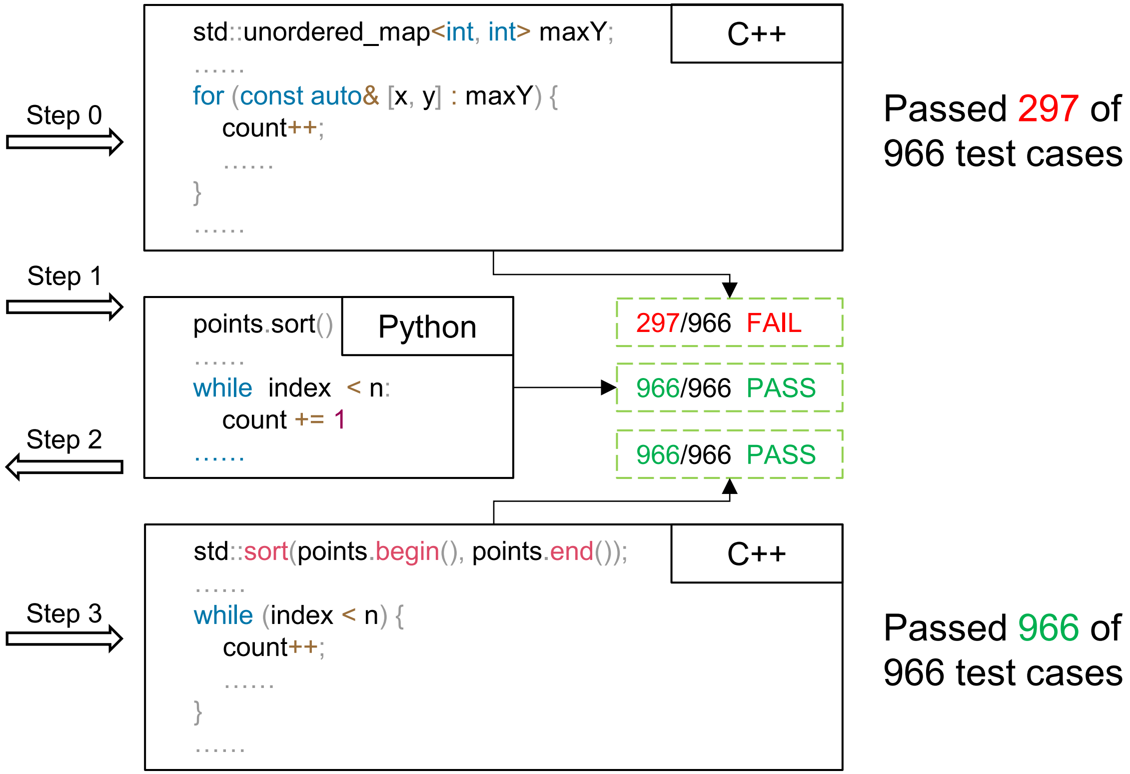
Each rectangle has its lower end at some point (x1, 0) and its upper end at some point (x2, y2), where x1 <= x2, y2 >= 0, and x2 - x1 <= w.

Summary from High-proficiency Language

The function first sorts the points based on their x-coordinates.

It then iterates through the points, placing a rectangle starting at the current point and extending to cover points within the width range.

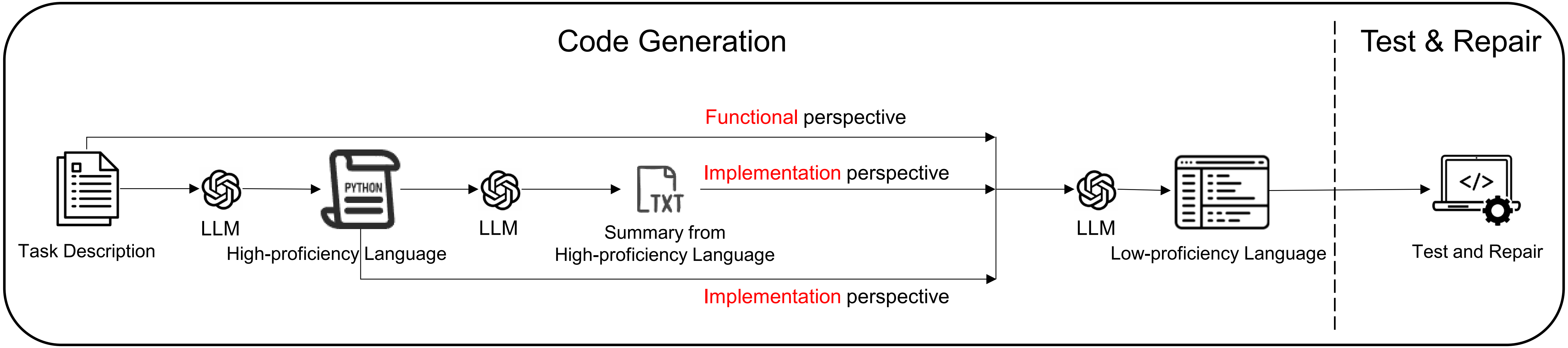
The loop continues until all points are covered, and the function returns the total number of rectangles used.



- **Functional perspective (**What does this task want to do?**)**: Task Description
- **Implementation perspective (**How to accomplish this task?**)**: Python Code and Summary

Workflow of MetaCoder

- **Code Generation**
  - Use high-proficiency language to guide the generation of low-proficiency language from multiple perspectives
- **Test & Repair**
  - Find out errors and repair



Code Generation

- **Provide information to LLMs in the form of dialogue**
  - Prompts are simple and add little workload to users
  - The process is easy and intuitive

Write a C++ code to solve the task:
Given two arrays `A` and `B`, find an integer x such that when each element in ` A ` is increased by x, ` A ` becomes equal to ` B `.

```
int fun(vector<int>& A, vector<int>& B) {
    return *min_element(begin(B), end(B)) - *min_element(begin(A), end(A));
}
```

Write a Python code to solve the task: Given two arrays `A` and `B`, find an integer x ···

```
def fun(A, B):
    return min(B) - min(A)
```

Summarize and analyze the Python code you generated.

The method returns the difference between the minimum value in `B` and the minimum value in `A`.

Write a C++ code to solve the task above.

```
int fun(vector<int>& A, vector<int>& B){···}
```

- **Why add summary to guide?**

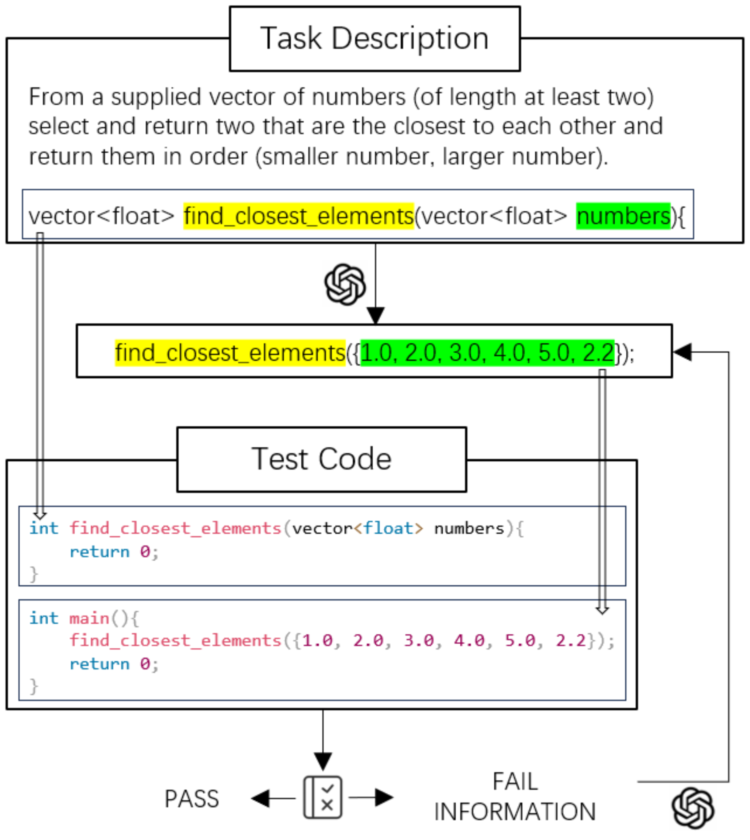
Syntactic gap between two different languages may lead to syntax errors

Test and Repair

LLMs make mistakes in function declarations:

Wrong function name、wrong parameter order and data type

- **Generate one test case without answer**
  - According to the function declaration
- **Test**
  - Compile to check errors
  - Collect fail information
- **Repair**
  - Add fail information to the dialogue
  - Generate and retest for several rounds



Experimental Design

- **Model selection**
  - GPT-3.5、GPT-4o、Llama 3.1、Qwen 2.5、DeepSeek V2.5、DeepSeek R1
- **Comparison Baselines**
  - Zero-Shot、Few-Shot、CoT、INTERVENOR、Self-Collaboration
- **Benchmarks**
  - HumanEval-x
- **Evaluation Metrics**
  - Pass@1 (5 samples for each task)
- **Research Questions**
  - RQ1: What is the effectiveness of our method?
  - RQ2: What is the effectiveness of each component of our method?
  - RQ3: What is the cost of our method?

Results

- **RQ1: What is the effectiveness of our method?**
  - Effectively improve the ability to generate low-proficiency languages

	GPT 3.5		GPT 4o		DeepSeek V2.5		Llama3.1 70B		Qwen2.5 72B	
	C++	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**
Zero-Shot	66.22	64.27	84.63	87.2	82.56	81.95	73.78	72.2	85.98	86.59
Few-Shot	60	67.68	80.98	85	85.37	83.54	78.05	78.65	86.58	85.37
CoT	64.51	66.34	84.39	85.73	**86.59**	**87.2**	76.22	**81.1**	85.98	86.58
INTERVENOR	66.95	71.95	84.02	88.41	83.41	80.73	71.95	74.27	88.41	85.98
Self-Collaboration	72.56	69.51	85.98	**90.85**	83.54	85.24	75.61	76.22	**90.24**	**89.02**
MetaCoder	**74.89**	**74.76**	**87.2**	85.49	**86.59**	86.34	**80**	78.65	86.58	87.8
Relative Improvement	13.09%↑	16.32%↑	3.04%↑	-1.96%↓	4.88%↑	5.36%↑	8.43%↑	8.93%↑	0.7%↑	1.4%↑

	Qwen2.5 32B		Qwen2.5 14B		Qwen2.5 7B		DeepSeek R1(C++)			
	C++	**Java**	**C++**	**Java**	**C++**	**Java**	**7B**	**8B**	**14B**	**32B**
Zero-Shot	78.9	74.02	72.2	77.44	68.29	69.39	47.93	50.61	54.88	53.05
MetaCoder	**84.76**	**86.59**	**73.54**	**79.63**	**71.95**	**71.59**	**63.41**	**62.8**	**80.49**	**81.1**
Relative Improvement	7.43%↑	16.98%↑	1.86%↑	2.83%↑	5.36%↑	3.17%↑	32.3%↑	24.09%↑	46.67%↑	52.87%↑

Results

- **RQ2: What is the effectiveness of each component of our method?**
  - Each component contributes to the improvement of ability
  - Summary and Python code complement each other and achieve better results
  - For more powerful LLMs, Test and Repair may provide less return

	GPT 3.5		GPT 4o		DeepSeek V2.5		Llama3.1 70B		Qwen2.5 72B	
	C++	Java	C++	Java	C++	Java	C++	Java	C++	Java
Zero-Shot	66.22	64.27	84.63	**87.2**	82.56	81.95	73.78	72.2	85.98	86.59
- Python Solution	73.66	74.02	85.98	85.61	84.76	85.98	77.93	78.17	85	86.84
- Python Summary	71.34	73.78	86.71	84.51	85.98	84.39	76.22	77.44	83.41	84.27
- Test and Repair	72.56	71.95	86.59	85.12	85	85.37	76.71	76.45	86.21	86.34
MetaCoder	**74.89**	**74.76**	**87.2**	85.49	**86.59**	**86.34**	**80**	**78.65**	**86.58**	**87.8**

Results

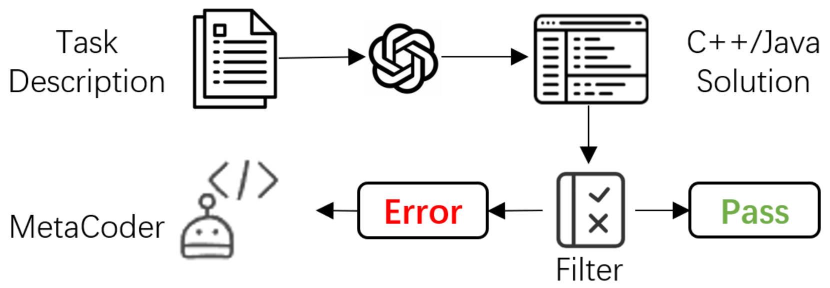
- **RQ3: What is the cost of our method?**
  - The cost of generating code has not increased significantly
  - There is a big difference in cost between various LLMs

The output token consumption of different LLMs under different methods

	GPT 3.5		GPT 4o		DeepSeek V2.5		Llama3.1 70B		Qwen2.5 72B	
	C++	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**
Zero-Shot	90.8	177.26	274.74	572.6	673.27	737.4	465.9	542.93	582.32	614.86
Few-Shot	109.35	113.54	171.44	130.39	195.15	167.01	144.62	118.43	253.71	136.36
CoT	302.41	327.21	499.53	545.56	507.34	540.45	364.76	370.88	568.46	407.27
INTERVENOR	99.23	190.32	292.62	611.46	681.23	815.26	510.07	545.45	614.49	627.62
Self-Collaboration	730.81	750.21	1231.91	1833.1	1606.19	1556.74	919.16	908.2	1697.06	1608.95
MetaCoder	396.61	421.35	690.47	824.3	738.02	848.67	438.99	459.13	637.67	609.41

Results

- **More detailed research: Real world performance**
  - After adding the filter, the effect is significantly improved
  - Enhance the diversity of LLM-generated code
  - Practical in the real world



	GPT 3.5		GPT 4o		DeepSeek V2.5		Llama3.1 70B		Qwen2.5 72B	
	C++	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**	**C++**	**Java**
MetaCoder	74.89	74.76	87.2	85.49	86.59	86.34	80	78.65	86.58	87.8
MetaCoder + Filter	81.59	81.22	90.73	91.22	90.85	90	86.58	86.58	91.46	92.07

	Qwen2.5 32B		Qwen2.5 14B		Qwen2.5 7B		DeepSeek R1(C++)			
	C++	**Java**	**C++**	**Java**	**C++**	**Java**	**7B**	**8B**	**14B**	**32B**
MetaCoder	84.76	86.59	73.54	79.63	71.95	71.59	63.41	62.8	80.49	81.1
MetaCoder + Filter	90.12	91.46	85.12	88.41	80.73	85.85	71.34	74.39	84.76	85.37

Thanks!

Github: **https://github.com/cx-hub/MetaCoder**

Contact: **chenxin19@nudt.edu.cn**