ELABORATION: A Comprehensive Benchmark on Human-LLM Competitive Programming

Xinwei Yang[1], Zhaofeng Liu2, Chen Huang1, Jiashuai Zhang1, Tong Zhang1, Yifan Zhang3, Wenqiang Lei1

[1]Sichuan University [2]Tianjin University of Science and Technology [3]Vanderbilt University

Human-LLM Competitive Programming

◉ **Background:**

1. While large language models have attracted attention in competitive programming, **their practical utility remains limited** due to underwhelming performance.
2. To address this, **Human-LLM Competitive Programming employs a human-in-the-loop approach**, using multi-turn feedback to improve LLM performance in competitive programming.

◉ **Motivation:**

- Existing research on Human-LLM collaboration in competitive programming is fragmented and **lacks a unified benchmark** to evaluate its effectiveness throughout the full problem-solving process.

◉ **Our Work:**

- We present **ELABORATION**, a novel benchmark for Human-LLM competitive programming with **a comprehensive evaluation protocol** that includes **a taxonomy of human feedback** and **a new human-LLM programming dataset** to assess the entire competitive programming process.

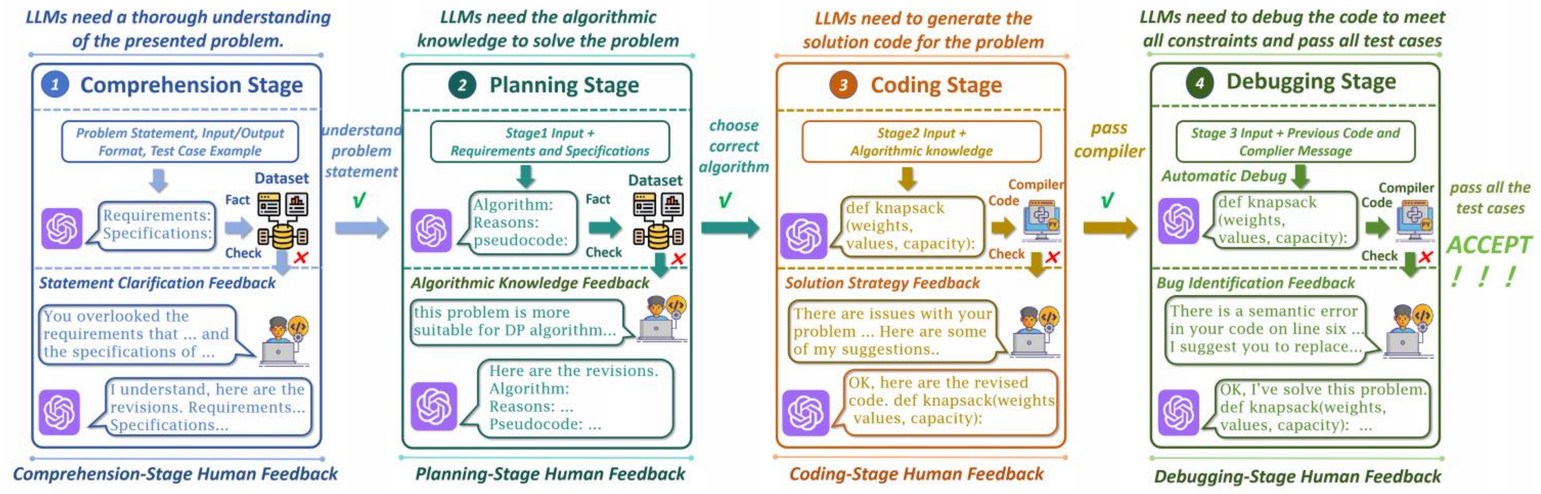
ELABORATIONSET:Benchmark Dataset

Dataset	Easy	Middle	Difficult
Basic Problem Information			
Time Period	Oct. 2011 ~ Nov. 2024		
#Problems	3642	2098	2580
Avg. #Test Cases	14.4	14.5	14.2
Annotations for Human Interaction (per Problem)			
Avg. #Statement Clarifications	8.1	10.9	12.1
Avg. #Algorithm Knowledge Summaries	2.4	3.0	3.8
Avg. #Ground Truth Solutions	4.8	4.9	4.8
Interaction Records with Real Humans			
#Problems	100	100	100
Avg. #Turns (#Human Feedback)	3.4	5.1	6.9
Avg. #Human-Annotated LLM Code Errors	1.3	1.5	2.0

Evaluation Protocol & Human Feedback Taxonomy

Evaluation Protocol

This diagram presents a four-stage evaluation framework for assessing LLMs in programming tasks — Comprehension, Planning, Coding, and Debugging.
At each stage, the model receives structured inputs and iterative human feedback to progressively refine its understanding, algorithm design, implementation, and final correctness, aiming to pass all test cases successfully.



Human Feedback Taxonomy

- **Problem Construction**
LLMs require a thorough understanding of the problem statement. To facilitate this, human feedback can provide crucial requirements and specifications.

- **Solution Planning**
LLMs plan solutions by selecting suitable algorithms, guided by human feedback offering suggestions, justifications, or pseudocode.

- **Code Generation**
LLMs must generate complete, compilable code. In this case, human feedback can suggest solution strategies to improve the generated code.

- **Code Debugging**
LLMs must pass the complete set of test cases. In this case, humans assist in identifying errors until all unseen test cases are passed.

Benchmark Analysis

Experimental Setup

◉ **Task & Data**: ELABORATIONSET
◉ **Baselines**: O1-Mini, GPT-4o, GPT-4-Turbo, Gemini-1.5-pro , Claude-3.5, CodeLlama-Series, Deepseek-Coder-Series, Qwen2.5-Coder-Series.
◉ **Evaluation Metrics**: we utilize the Pass@k (k=1,3,5)5 metric (Chen et al., 2021) to evaluate overall performance.

◉ **Overall Performance (RO1) :**

1. They experimental results demonstrate **limited capacity** for solving competitive programming problems, particularly those of **high difficulty or unseen ones**.
2. Human-LLM collaboration **significantly enhances** LLM performance, demonstrating the crucial role of human feedback.

◉ **Finer-grained Analysis(RQ2):**

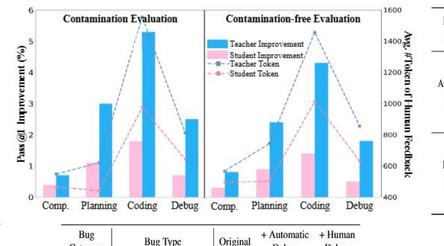
1. During **the coding stage**, even on problems with no data contamination, the human feedback is most beneficial.
2. Expert feedback(Teacher Programmer) yields **greater benefits**, its higher cost necessitates efficient use of human resources.

Model (Cut-off Date|Release Date)	Contamination Evaluation (%)				Contamination-free Evaluation (%)			
	Easy	Middle	Hard	Overall	Easy	Middle	Hard	Overall
O1-Mini (2023-12 | 2024-09)	88.1	70.3	41.7	66.7	80.6	66.6	30.8	59.3
GPT-4o (2023-11 | 2024-05)	80.4	50.5	20.8	50.6	74.1	31.7	10.3	38.7
+ Student Programmer Feedback	83.1	53.1	24.3	53.5	76.2	34.8	15.1	42.0
+ Teacher Programmer Feedback	87.7	66.1	38.2	64.0	80.1	42.9	23.3	48.8
GPT-4-Turbo (2023-05 | 2023-11)	70.5	40.6	8.7	39.9	65.2	27.3	5.8	32.8
+ Student Programmer Feedback	75.5	46.1	12.1	44.6	70.8	33.2	8.8	37.6
+ Teacher Programmer Feedback	83.2	58.8	20.1	54.0	75.3	39.8	14.3	43.1
Gemini-1.5-pro (2023-11 | 2024-02)	81.2	48.2	22.0	50.5	73.2	32.8	9.3	38.4
+ Student Programmer Feedback	84.0	50.1	25.1	53.0	75.5	35.0	13.1	41.2
+ Teacher Programmer Feedback	89.1	65.6	36.6	63.8	81.0	40.2	24.2	48.5
Claude-3.5 (2024-03 | 2024-06)	78.0	51.3	16.2	48.5	74.5	34.3	5.4	38.1
+ Student Programmer Feedback	82.2	55.0	24.1	53.8	76.6	37.1	7.9	40.5
+ Teacher Programmer Feedback	87.0	66.7	33.4	62.4	83.1	44.2	16.5	47.9
Avg.	77.5	47.7	16.9	47.4	71.8	31.5	7.7	37.0
+ Student Programmer Feedback	81.2 (+3.7)	51.1 (+3.4)	21.4 (+4.5)	51.2 (+3.8)	74.8 (+3.0)	35.0 (+3.5)	11.2 (+3.5)	40.3 (+3.3)
+ Teacher Programmer Feedback	86.8 (+9.3)	64.3 (+16.6)	32.1 (+15.2)	61.1 (+13.7)	79.9 (+8.1)	41.8 (+10.3)	19.6 (+11.9)	47.1 (+10.1)
~7B Scale								
CodeLlama-7B (2023-01 | 2024-01)	30.3	5.9	0.5	12.2	15.2	2.1	0.3	5.9
+ Student Programmer Feedback	36.7	10.3	2.2	16.4	24.2	3.1	1.4	9.6
+ Teacher Programmer Feedback	48.6	17.8	6.9	24.4	35.9	8.4	4.7	16.3
Deepseek-Coder-6.7B (2023-09 | 2023-11)	40.6	15.4	1.8	19.3	21.4	7.0	0.7	9.7
+ Student Programmer Feedback	46.3	18.8	4.3	23.1	27.8	11.3	2.0	13.7
+ Teacher Programmer Feedback	58.6	27.8	8.2	31.5	39.2	24.2	6.1	23.2
Qwen2.5-Coder-7B (2024-06 | 2024-11)	61.2	22.4	4.9	29.5	48.6	9.3	0.5	19.5
+ Student Programmer Feedback	70.1	26.6	6.7	34.5	53.8	12.3	2.3	22.8
+ Teacher Programmer Feedback	76.3	35.5	11.3	41.0	57.8	21.6	5.9	28.4
Avg.	44.0	14.6	2.4	20.3	28.4	6.1	0.5	11.7
+ Student Programmer Feedback	51.0(+7.0)	18.6(+4.0)	4.4(+2.0)	24.7(+4.4)	35.3(+6.9)	8.9(+2.8)	1.9(+1.4)	15.4(+3.7)
+ Teacher Programmer Feedback	61.2(+17.2)	27.0(+12.4)	8.8(+6.4)	32.3(+12.0)	44.3(+15.9)	18.1(+12.0)	5.6(+5.1)	22.6(+10.9)
~13B Scale								
CodeLlama-13B (2023-01 | 2024-01)	35.8	7.3	1.7	14.9	23.5	3.0	0.3	8.9
+ Student Programmer Feedback	40.3	12.1	2.9	18.4	26.3	9.8	1.4	12.5
+ Teacher Programmer Feedback	44.2	19.9	5.8	23.3	29.8	14.6	3.1	15.8
Qwen2.5-Coder-14B (2024-06 | 2024-11)	70.8	28.7	7.7	35.7	58.3	15.1	2.2	25.2
+ Student Programmer Feedback	75.9	33.5	10.2	40.0	61.2	18.9	4.1	28.1
+ Teacher Programmer Feedback	80.1	41.5	14.2	45.3	66.3	24.3	6.8	32.5
Avg.	53.3	18.0	4.7	25.3	40.9	9.1	1.3	17.1
+ Student Programmer Feedback	58.1 (+4.8)	22.8 (+4.8)	6.6 (+1.9)	29.2 (+3.9)	43.8 (+2.9)	14.4 (+5.3)	2.8 (+1.5)	20.3 (+3.2)
+ Teacher Programmer Feedback	62.2 (+8.9)	30.7 (+12.7)	10.0 (+5.3)	34.3 (+9.0)	48.1 (+7.2)	19.5 (+10.4)	5.0 (+3.7)	24.2 (+7.1)
~34B Scale								
CodeLlama-34B (2023-01 | 2024-01)	38.1	7.9	3.1	16.4	25.0	5.1	1.0	10.4
+ Student Programmer Feedback	42.0	12.3	4.0	19.4	26.1	8.4	2.3	12.3
+ Teacher Programmer Feedback	49.2	18.8	6.2	24.7	32.2	13.0	4.4	16.5
Deepseek-Coder-33B (2023-09 | 2023-11)	63.9	23.7	4.2	30.6	50.6	10.4	1.2	20.7
+ Student Programmer Feedback	74.8	28.7	7.0	36.8	55.8	13.3	3.1	24.0
+ Teacher Programmer Feedback	78.9	40.1	12.3	43.8	68.8	20.4	5.5	31.6
Qwen2.5-Coder-32B (2024-06 | 2024-11)	77.3	41.3	9.0	42.5	70.1	20.3	3.2	31.2
+ Student Programmer Feedback	80.4	45.3	11.0	45.6	72.0	23.1	4.0	33.0
+ Teacher Programmer Feedback	85.1	53.4	15.8	51.4	76.8	30.1	7.6	38.0
Avg.	59.8	24.3	5.4	29.8	48.6	11.9	1.8	20.8
+ Student Programmer Feedback	65.7(+5.9)	28.8(+4.5)	7.3(+1.9)	33.9(+4.1)	51.3(+2.7)	14.9(+3.0)	3.1(+1.3)	23.1(+2.3)
+ Teacher Programmer Feedback	71.1(+11.3)	37.4(+13.1)	11.4(+6.0)	40.0(+10.2)	59.3(+10.7)	21.2(+9.3)	5.8(+4.0)	28.7(+7.9)
Average over All LLMs	60.7	28.6	8.4	32.6	50.0	16.5	3.4	23.3
+ Student Programmer Feedback	65.9(+5.2)	32.7(+4.1)	11.2(+2.8)	36.6(+4.0)	53.9(+3.9)	20.0(+3.5)	5.5(+2.1)	26.4(+3.1)
+ Teacher Programmer Feedback	72.3(+11.6)	42.7(+14.1)	17.4(+9.0)	44.1(+11.5)	60.5(+10.5)	27.0(+10.5)	10.2(+6.8)	32.6(+9.3)

Student Programmer: (Intermediate Skill Level) possess more than basic programming knowledge but lack the deep expertise.
Teacher Programmer: (Expert Level) possess a high level of programming skill and experience.

◉ **Collaborating with Real Humans(RQ3):**

1. Humans **play a vital role** in identifying bugs and improving LLM performance.
2. Human and LLMs **have complementary strengths**, creating a powerful synergy.



Debug Type	Difficulty Level	Error Identification: Precision	Error Identification: Recall	Problem Resolution (P@1): Original	Problem Resolution (P@1): + Debug
Automatic Debug	Easy	0.34	0.56	0.66	0.73
	Middle	0.22	0.36	0.27	0.33
	Hard	0.14	0.28	0.06	0.08
	Overall	0.23	0.40	0.33	0.38
Human Debug	Easy	0.92	0.78	0.73	0.92
	Middle	0.80	0.72	0.33	0.65
	Hard	0.72	0.64	0.08	0.29
	Overall	0.81	0.71	0.38	0.62

Bug Category	Bug Type	Original	+ Automatic Debug	+ Human Debug
Syntactic Bugs	Function Related Errors	11	4	1
	Operation Errors	3	1	0
	Structure Errors	4	1	0
	Declaration Errors	8	2	0
	Import Errors	7	2	0
	Overall	33	10	1
Semantic Bugs	Control Flow Errors	58	46	30
	Reference Errors	17	16	3
	Calculation Errors	23	25	4
	Incomplete Errors	142	99	55
	Logical Direction Error	87	33	12
	Suboptimal Errors	23	20	19
	Overall	350	239	123