Can Multi-turn Self-refined Single Agent LMs with Retrieval Solve Hard Coding Problems?

Md Tanzib Hosain[1,3,*] Md Kishor Morol[2,3]

[1]American International University-Bangladesh [2]Cornell University [3]EliteLab.AI
[*]Work done while working as a remote RA at QCRI.
20-42737-1@student.aiub.edu, mmorol@cornell.edu

ACL 2025 VIENNA

Motivation

- Widely used benchmarks for code generation, such as HumanEval and MBPP, have become saturated, with language models achieving solve rates exceeding 90%. This necessitates more challenging benchmarks to accurately assess and differentiate the capabilities of state-of-the-art models.
- Prior research on competitive programming for LMs has often been hindered by a lack of comprehensive unit test suites, official problem analyses, or a sufficient variety of problems to thoroughly evaluate algorithmic reasoning.
- Even a top-performing model like "o1" achieves only a 19.1% pass@1 solve rate using a standard zero-shot chain-of-thought approach, indicating a clear need for more advanced techniques.
- The research seeks to introduce more difficult benchmarks and inference techniques that expose the shortcomings of current models.

Dataset Statistics

Selected Problems# in Contest Venues

Category	Problems#
World Final & Continent Finals	167
Regionals	87
Total	**254**

Methodology

Multi-turn Self-judge Framework

Architecture Overview: Problem → LLM → Knowledge Retrieval → Self-judge → Feedback Loop

Problem Sources

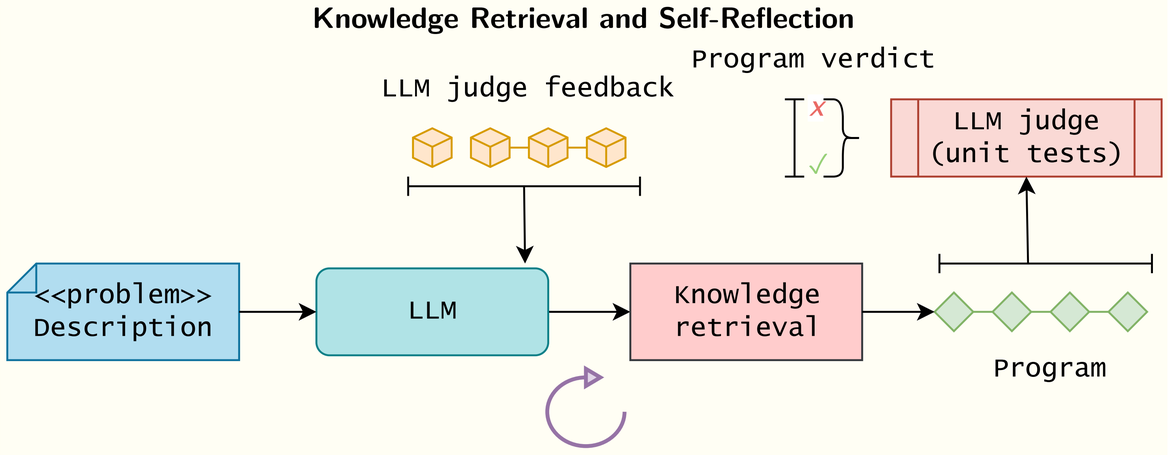
- **Episodic Retrieval**: Similar problems with solutions
- **Self-judge**: Unit test validation
- **Self-Reflection**: Learning from execution feedback
- **Multi-turn Iteration**: Iterative refinement

Inference Techniques

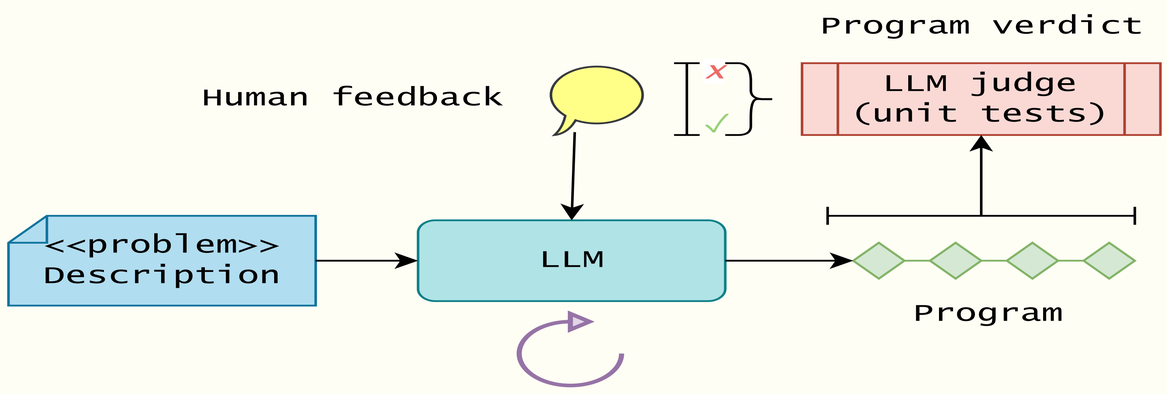
- Zero-shot Chain-of-Thought
- Few-shot prompting
- Semantic & Episodic retrieval
- Self-reflection
- Combined approaches

Framework Architecture

Knowledge Retrieval and Self-Reflection



Human Agent Interaction



Results/Findings

Zero-shot Pass@1 Model Performance

Model	Pass@1
gpt-4	7.3
claude-3.5-sonnet	14.1
gpt-4o	14.2
qwen2.5-coder	14.8
athene-v2-chat	16.4
deepSeek-v3-chat	17.6
gemini-exp	18.3
o1	**19.1**

Inference Method's Pass@1 Performance

Inference technique	Model		
	gpt-4	gpt-4o	o1
zero_shot	7.3	14.2	19.1
brainstorm_then_select	8.6	16.9	21.7
few_shot	10.1	19.4	24.2
self_reflection	11.3	20.6	25.4
semantic_retrieval	12.4	22.1	27.3
semantic_retrieval + self_reflection	12.8	22.5	28.1
episodic_retrieval	13.2	23.3	29.0
semantic_retrieval + episodic_retrieval	14.5	24.4	29.8
semantic_retrieval + episodic_retrieval + self_reflection	16.4	27.1	33.2
episodic_retrieval + self_reflection	**24.3**	**38.4**	**42.2**

Integrating Human Feedback

Interaction Rules

- **Allowed**: General concepts, sample walkthrough, high-level directions
- **Forbidden**: Exact algorithms, specific code fixes, detailed explanations

Final %Solve (Participants in this interaction module have Codeforces rating > 2500.)

Model	Final solve rate
gpt-4	0
gpt-4o	0
o1	0
o1 + interact	**94.4**

Error Analysis

Episodic Retrieval + Self-Reflection %Errors

Model	Wrong Ans.	TLE	MLE	Runtime	Syntax + Other
gpt-4	58.81	5.33	0	10.16	1.38
gpt-4o	28.95	25.06	0	6.83	0.77
o1	27.87	23.56	0	5.78	0.59

Ablation Studies

Retrieval Query Ablation Performance

Query	Pass@1
problem_description	28.5
problem_description + proposed_code_solution	**29.0**
problem_description + proposed_solution + code_solution	29.8

Hyperparameter Tuning on Problems# Retrieve for EPisodic Retrieval

Problems	Pass@1
$p = 1$	28.1
p **= 2**	**29.0**
$p = 3$	28.4

Iteration Tuning of "o1" Model on Iterations# for Self-Reflection

Iterations	Pass@1
$i = 0$	21.3
$i = 1$	23.8
i **= 2**	**25.6**
$i = 3$	25.4