



Graph Reasoning via Self-Correcting Multi-Agent Debate

Group: WW



Introduction

Large Language Models (LLMs) struggle with tasks requiring **structured algorithmic reasoning**, such as solving graph-based problems.

Describing a graph textually is possible, but reliably executing complex algorithms remains an open challenge. The goal of this project is to **dramatically improve** the reliability and accuracy of LLMs in this domain.

GraphWiz: An Instruction-Following Language Model for Graph Computational Problems



GraphWiz

- Finetuning of an open-source LLM in solving computational graph problems.
- It generates step-by-step reasoning paths for its solutions.
- Trained on a custom dataset named GraphInstruct.



Issues:

The GraphWiz model generates a **single "chain of thought"**. This process is monolithic: an early error compromises the entire solution, making it brittle.

The background is a dark blue gradient with abstract white geometric patterns. A large wireframe sphere with a central void is positioned in the upper left. Below it, a wireframe shape resembling a hand or a series of connected points is visible. Various small triangles and lines are scattered across the scene. On the right, a dark blue rectangular box with a white border contains text. Above this box, there are several white geometric shapes: a hexagon, a pentagon, and a square, all outlined in white.

Our Solution: A Multi-Agent Debate Framework

We propose to replace the single chain with a **dynamic debate** between agents that propose, verify, and correct steps, making the process robust and self-correcting.



Proposed System Architecture: Agent Roles

Proposer Agent

Generates a single logical step towards the solution (e.g., "From node X, I propose moving to node Y").

Verifier (Critic) Agent

Analyzes the proposal, validates it against the rules, and provides feedback (e.g., "Correct" or "Error: node already visited").

Coordinator Agent

Oversees the debate, interprets feedback, and prompts the Proposer for a new move in case of an error.



Case study



Challenge

To find a Hamiltonian path that visits each node exactly once.



Process

1. **Proposer:** "From node C, I propose node A."
2. **Verifier:** "Error. Node A has already been visited."
3. **Coordinator:** "Proposal rejected. Try again from node C."



Solution

A self-correction loop that leads to a valid solution.

Timeline



A

Baseline

We will implement the framework and test it on the *GraphInstruct* dataset to compare accuracy with GraphWiz.

B

Ablation Studies

We will analyze the impact of different configurations (number of agents, agent and LLM type) on performance.

C

Generalization

We will test the system on other graph datasets to evaluate its generalization capabilities.

D

Extension

We will integrate external tools to enhance the Verifier Agent.



Reviews



The evaluation will be purely quantitative and objective, performed on the *GraphInstruct* benchmark.

Results & Testing Process



The primary metric will be **Solution Accuracy**, calculated by comparing our system's answer with the ground truth.

Metrics



Success is defined as statistically surpassing the accuracy reported by the original GraphWiz model.

Success Criterion





Table of contents



01

Framework

A novel multi-agent framework for graph reasoning.

02

Performance

An empirical demonstration of improved accuracy and robustness.



03

Analysis

An analysis of how different agent configurations affect performance.

04

Innovation

A step towards more reliable LLMs for complex procedural problems.

05

Conclusion

In summary, our work aims to make AI more trustworthy.

The background features a dark blue gradient with abstract white circuit-like lines and shapes. These include straight lines, right-angle turns, circles, and various polygons (hexagons, octagons, and squares) that resemble components on a circuit board. Some shapes are solid white, while others are outlines. The overall aesthetic is modern and technological.

Thanks!

Group: WW