Final Project - Multi Agent Debate

Chess with LLMs

Project Introduction

The aim of the project is to look at self correction in LLMs and how multi-agent debate can drive the LLM to produce the correct solution.

Multi-agent debate is a framework through which multiple LLMs come together to discuss and provide the correct next logical step.

Project Description

In general, LLMs are not good at playing chess. Chess involves playing a move based on all the previous moves of both parties. LLMs perform well upto certain moves for well known openings in chess. But once it encounters moves which are not documented as such, it starts producing illegal moves. Our aim is to correct this by the use of Multi-agent debate. Two or more LLMs debate amongst themselves to provide a valid next move.

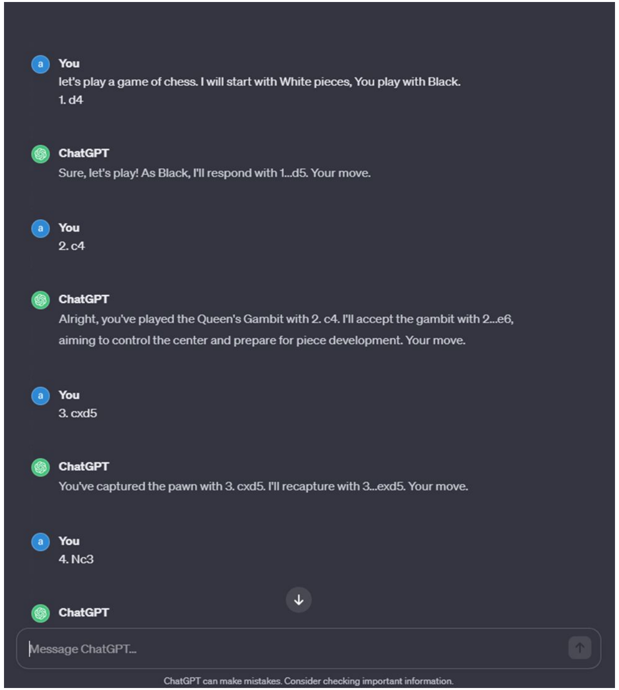
Approach

Our approach involved assessing the results across three distinct scenarios:

1. User vs Player(Single LLM instance)
2. User vs Player(First LLM) + Helper(Second LLM)
3. User vs Instance1(First LLM) + Instance2(Second LLM) + Judge
4. User vs Instance1(First LLM) + Instance2(Second LLM) + debate in 2 iterations + Judge

(Here, user indicates human playing against the LLMs.)

Execution-1 User vs Player(Single LLM instance)



You
let's play a game of chess. I will start with White pieces, You play with Black.
1. d4

ChatGPT
Sure, let's play! As Black, I'll respond with 1...d5. Your move.

You
2. c4

ChatGPT
Alright, you've played the Queen's Gambit with 2. c4. I'll accept the gambit with 2...e6, aiming to control the center and prepare for piece development. Your move.

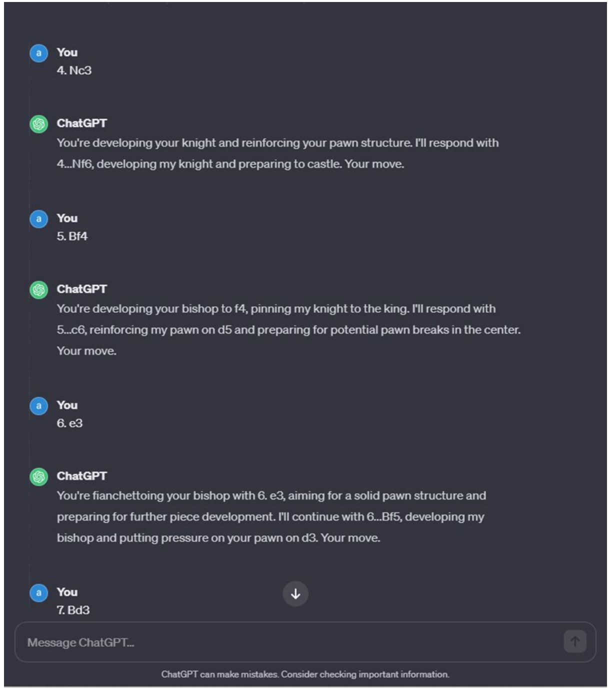
You
3. cxd5

ChatGPT
You've captured the pawn with 3. cxd5. I'll recapture with 3...exd5. Your move.

You
4. Nc3

ChatGPT

Message ChatGPT...
ChatGPT can make mistakes. Consider checking important information.



You
4. Nc3

ChatGPT
You're developing your knight and reinforcing your pawn structure. I'll respond with 4...Nf6, developing my knight and preparing to castle. Your move.

You
5. Bf4

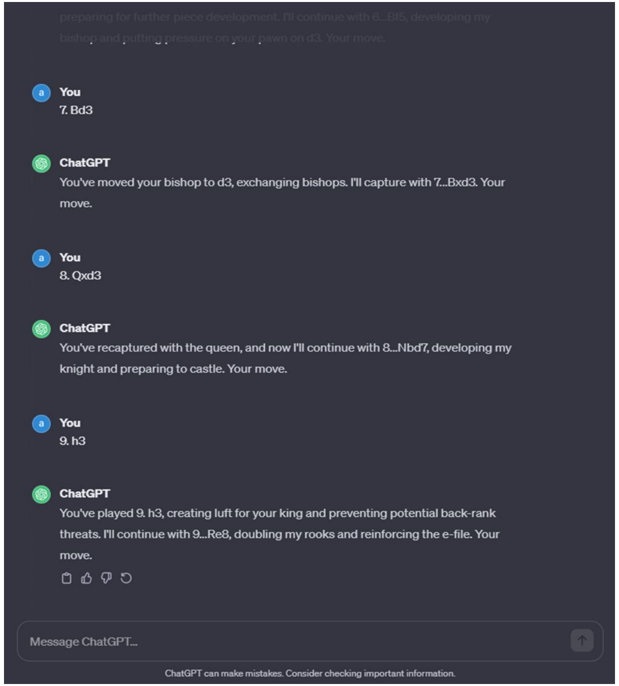
ChatGPT
You're developing your bishop to f4, pinning my knight to the king. I'll respond with 5...c6, reinforcing my pawn on d5 and preparing for potential pawn breaks in the center. Your move.

You
6. e3

ChatGPT
You're fianchettoing your bishop with 6. e3, aiming for a solid pawn structure and preparing for further piece development. I'll continue with 6...Bf5, developing my bishop and putting pressure on your pawn on d3. Your move.

You
7. Bd3

Message ChatGPT...
ChatGPT can make mistakes. Consider checking important information.



preparing for further piece development. I'll continue with 6...Bf5, developing my bishop and putting pressure on your pawn on d3. Your move.

You
7. Bd3

ChatGPT
You've moved your bishop to d3, exchanging bishops. I'll capture with 7...Bxd3. Your move.

You
8. Qxd3

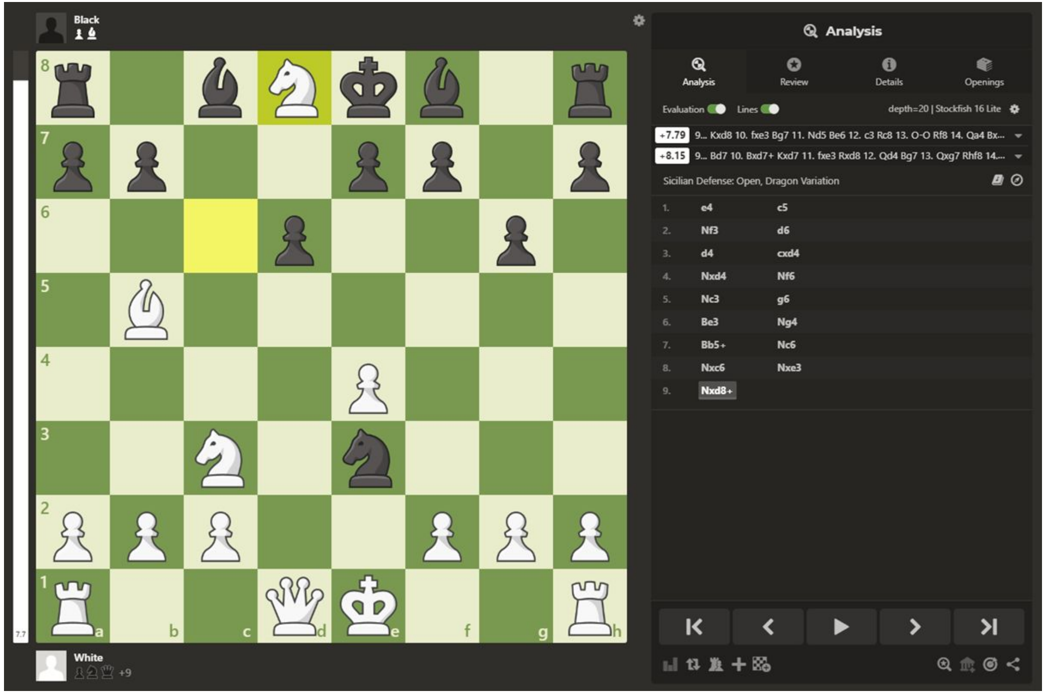
ChatGPT
You've recaptured with the queen, and now I'll continue with 8...Nbd7, developing my knight and preparing to castle. Your move.

You
9. h3

ChatGPT
You've played 9. h3, creating luft for your king and preventing potential back-rank threats. I'll continue with 9...Re8, doubling my rooks and reinforcing the e-file. Your move.

Message ChatGPT...
ChatGPT can make mistakes. Consider checking important information.

Execution-2 `User vs Player(First LLM) + Helper(Second LLM)`



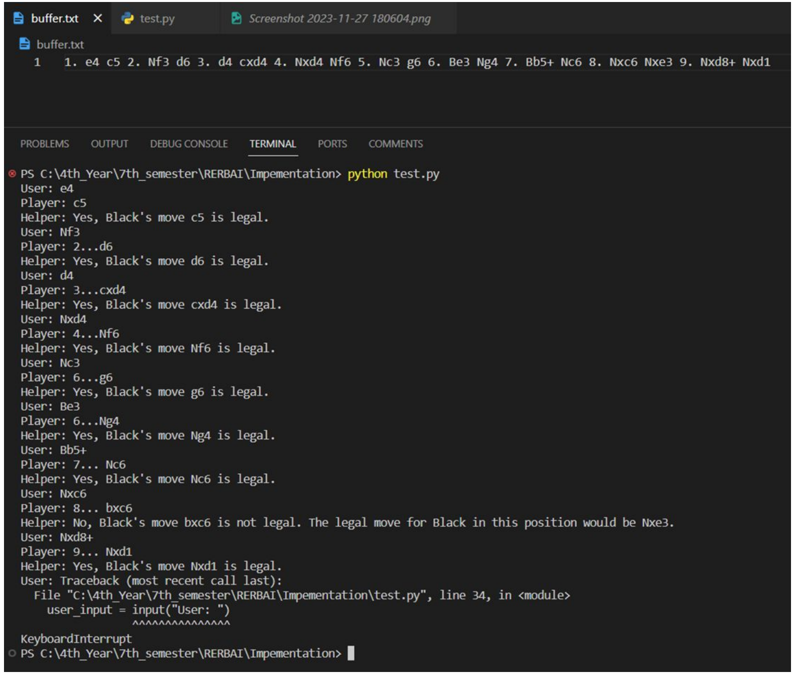
depth=20 | Stockfish 16 Lite

+7.79 9... Kxd8 10. fxe3 Bg7 11. Nd5 Be6 12. c3 Rc8 13. O-O Rf8 14. Qa4 Bx...

+8.15 9... Bd7 10. Bxd7+ Kxd7 11. fxe3 Rxd8 12. Qd4 Bg7 13. Qxg7 Rhf8 14....

Sicilian Defense: Open, Dragon Variation

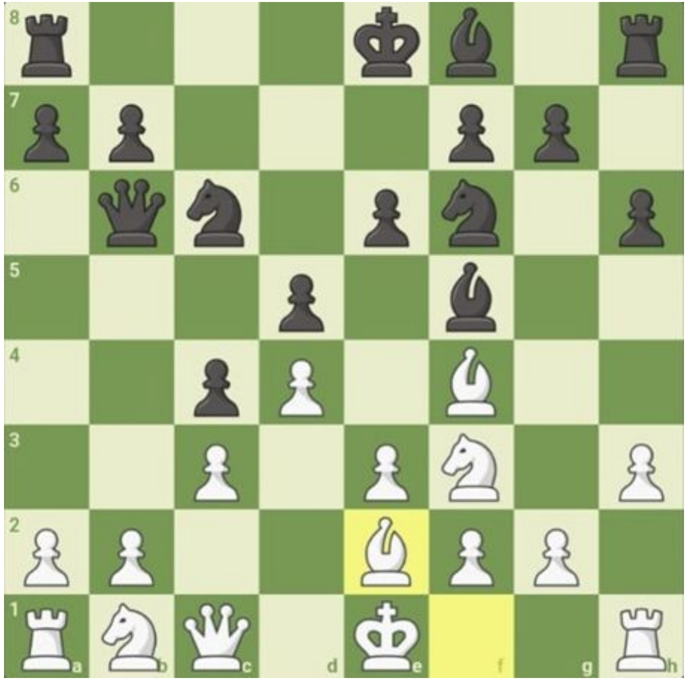
1.	e4	c5
2.	Nf3	d6
3.	d4	cxd4
4.	Nxd4	Nf6
5.	Nc3	g6
6.	Be3	Ng4
7.	Bb5+	Nc6
8.	Nxc6	Nxe3
9.	Nxd8+	

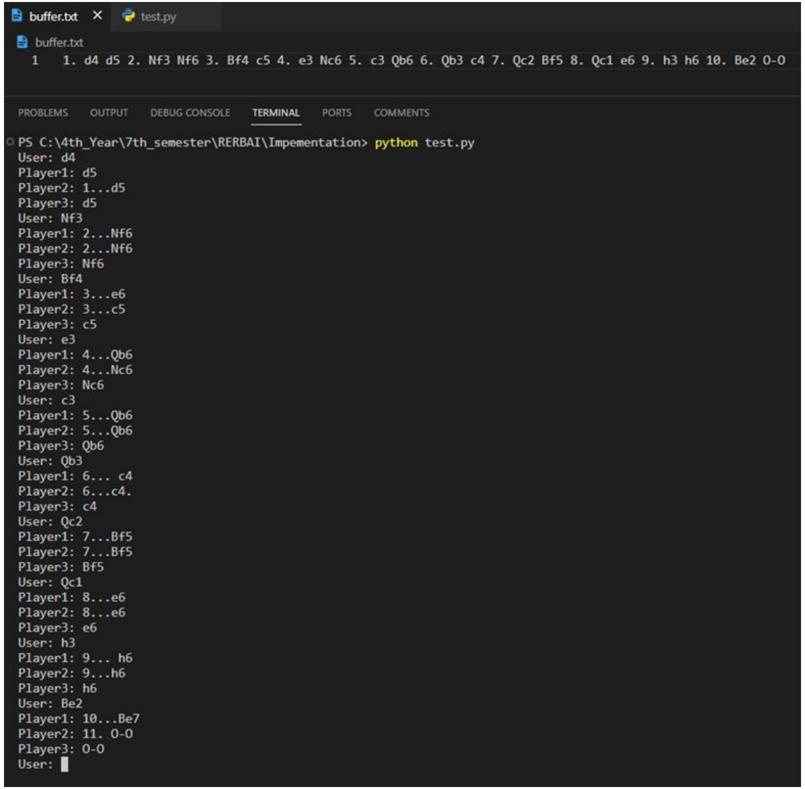


```
1. e4 c5 2. Nf3 d6 3. d4 cxd4 4. Nxd4 Nf6 5. Nc3 g6 6. Be3 Ng4 7. Bb5+ Nc6 8. Nxc6 Nxe3 9. Nxd8+ Nxd1
```

```
PS C:\4th_Year\7th_semester\RERBAI\Impementation> python test.py
User: e4
Player: c5
Helper: Yes, Black's move c5 is legal.
User: Nf3
Player: 2...d6
Helper: Yes, Black's move d6 is legal.
User: d4
Player: 3...cxd4
Helper: Yes, Black's move cxd4 is legal.
User: Nxd4
Player: 4...Nf6
Helper: Yes, Black's move Nf6 is legal.
User: Nc3
Player: 6...g6
Helper: Yes, Black's move g6 is legal.
User: Be3
Player: 6...Ng4
Helper: Yes, Black's move Ng4 is legal.
User: Bb5+
Player: 7... Nc6
Helper: Yes, Black's move Nc6 is legal.
User: Nxc6
Player: 8... bxc6
Helper: No, Black's move bxc6 is not legal. The legal move for Black in this position would be Nxe3.
User: Nxd8+
Player: 9... Nxd1
Helper: Yes, Black's move Nxd1 is legal.
User: Traceback (most recent call last):
  File "C:\4th_Year\7th_semester\RERBAI\Impementation\test.py", line 34, in <module>
    user_input = input("User: ")
                 ^^^^^^^^^^^^^^^
KeyboardInterrupt
PS C:\4th_Year\7th_semester\RERBAI\Impementation> 
```

Execution-3 User vs Instance1(1st LLM) + Instance2(2nd LLM) + Judge

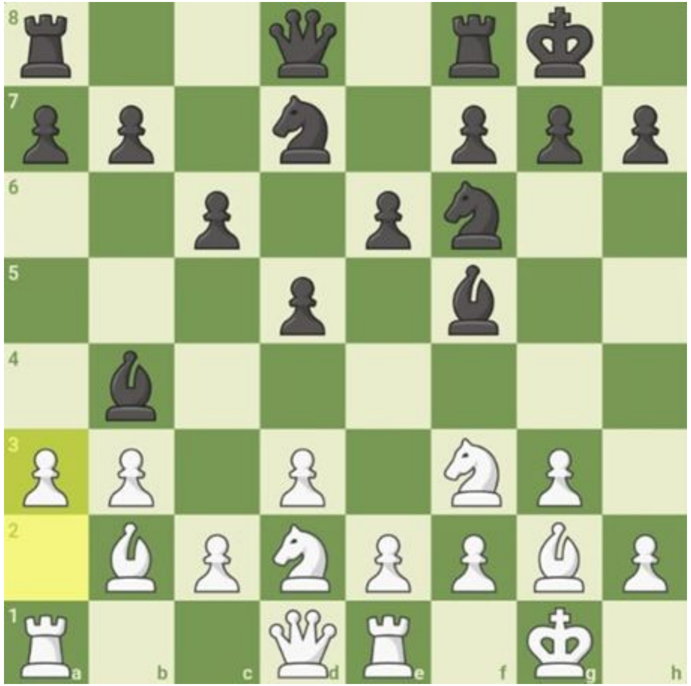


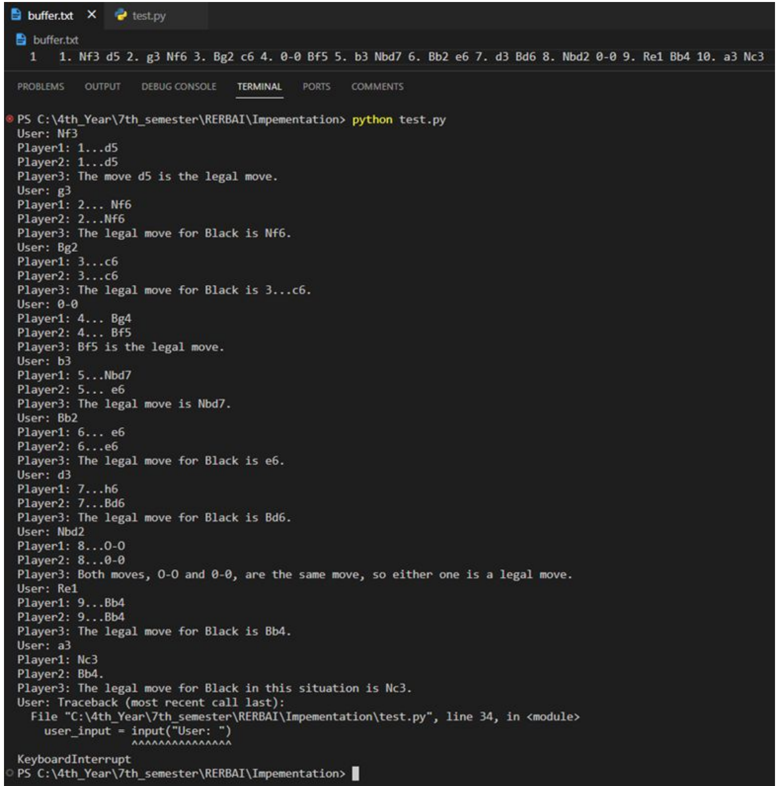


```
1. d4 d5 2. Nf3 Nf6 3. Bf4 c5 4. e3 Nc6 5. c3 Qb6 6. Qb3 c4 7. Qc2 Bf5 8. Qc1 e6 9. h3 h6 10. Be2 O-O
```

```
PS C:\4th_Year\7th_semester\RERBAI\Impementation> python test.py
User: d4
Player1: d5
Player2: 1...d5
Player3: d5
User: Nf3
Player1: 2...Nf6
Player2: 2...Nf6
Player3: Nf6
User: Bf4
Player1: 3...e6
Player2: 3...c5
Player3: c5
User: e3
Player1: 4...Qb6
Player2: 4...Nc6
Player3: Nc6
User: c3
Player1: 5...Qb6
Player2: 5...Qb6
Player3: Qb6
User: Qb3
Player1: 6... c4
Player2: 6...c4.
Player3: c4
User: Qc2
Player1: 7...Bf5
Player2: 7...Bf5
Player3: Bf5
User: Qc1
Player1: 8...e6
Player2: 8...e6
Player3: e6
User: h3
Player1: 9... h6
Player2: 9...h6
Player3: h6
User: Be2
Player1: 10...Be7
Player2: 11. O-O
Player3: O-O
User: 
```

Execution-3 User vs Instance1(1st LLM) + Instance2(2nd LLM) + Judge





```
1. Nf3 d5 2. g3 Nf6 3. Bg2 c6 4. 0-0 Bf5 5. b3 Nbd7 6. Bb2 e6 7. d3 Bd6 8. Nbd2 0-0 9. Re1 Bb4 10. a3 Nc3
```

```
PS C:\4th_Year\7th_semester\RERBAI\Impementation> python test.py
User: Nf3
Player1: 1...d5
Player2: 1...d5
Player3: The move d5 is the legal move.
User: g3
Player1: 2... Nf6
Player2: 2...Nf6
Player3: The legal move for Black is Nf6.
User: Bg2
Player1: 3...c6
Player2: 3...c6
Player3: The legal move for Black is 3...c6.
User: 0-0
Player1: 4... Bg4
Player2: 4... Bf5
Player3: Bf5 is the legal move.
User: b3
Player1: 5...Nbd7
Player2: 5... e6
Player3: The legal move is Nbd7.
User: Bb2
Player1: 6... e6
Player2: 6...e6
Player3: The legal move for Black is e6.
User: d3
Player1: 7...h6
Player2: 7...Bd6
Player3: The legal move for Black is Bd6.
User: Nbd2
Player1: 8...O-O
Player2: 8...0-0
Player3: Both moves, O-O and 0-0, are the same move, so either one is a legal move.
User: Re1
Player1: 9...Bb4
Player2: 9...Bb4
Player3: The legal move for Black is Bb4.
User: a3
Player1: Nc3
Player2: Bb4.
Player3: The legal move for Black in this situation is Nc3.
User: Traceback (most recent call last):
  File "C:\4th_Year\7th_semester\RERBAI\Impementation\test.py", line 34, in <module>
    user_input = input("User: ")
                 ^^^^^^^^^^^^^^^
KeyboardInterrupt
PS C:\4th_Year\7th_semester\RERBAI\Impementation>
```

Analysis

1. The Execution 2 and 3 did not particularly improve the results compared to the base model.
2. The performance in every scenario is heavily influenced on prompt engineering.
3. In case of multiple opinions, the judge seems to get confused and finds it difficult to deduce the correct move.
4. “Typical” Moves are Playable because they are not interrupted by the opponent.
5. If sufficient tokens are passed, MAD Framework improves the performance with 3-4 more moves but the reasoning might not be correct.
6. Luck plays some role!

Challenges faced

---

1. LLMs has by-hearted answers, not understood it. (In context of chess.)
2. Using LLM APIs had its own set of challenges -
   1. Delays
   2. Conversation History
   3. APIs are not free!
3. Token Limitations for the input/output prompts.