Self-Consistency Enhanced Chess Agents

CS395T - Allen Jue and Sampoorna Poria

Testbed

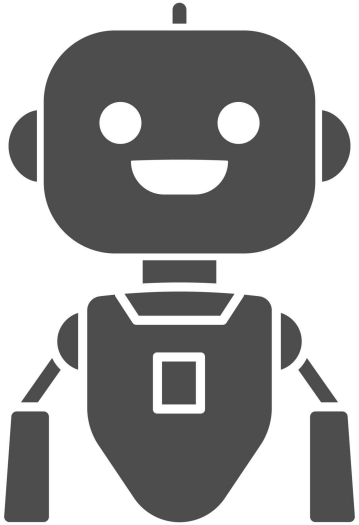
- Want to see if models can play puzzles, which require thinking multiple moves ahead.
- Sample 1000 Lichess puzzles with ratings from 600-2400.
- Tested a single model with this prompt, self-consistency, and modified the MAD code.

Project Overview and Motivation

- Chess is great for evaluating reasoning and planning
- It is already played at a superhuman level[1]
- Millions of people still play chess and many underexplored problems:
  - Plan-coherence and long-term strategy (not just find one best move)
  - How do different models perform?
  - Human-like play for education[2]

Related Work

- Multi-agent debate encourages fact-checking and deeper reasoning[3][4]
- Relatively few full chess playing LLMs[5] and no comprehensive LLM evaluations



Experimental Setup

- Prompt models with conventional chess prompts.
- Sampled 1,000 Lichess puzzles and tested on a subset of 50 puzzles.
- Tested on 9 models from various architectures and sizes
- Experimental Paradigms
    - Single response (baseline), Self-Consistency (SC), Multi-Agent Debate (MAD)

Results: Puzzle Performance

Model	Single	SC	MAD
Afm 4.5B	0.00	0.00	0.00
Deepseek V3	8.00	8.00	8.00
Gemma 2 2B It	0.00	0.00	0.00
Gpt 3.5 Turbo Instruct	*44.00*	**54.00**	46.00
Llama 3.1 8B Instruct	0.00	0.00	0.00
Llama 3.3 70B Instruct	2.00	4.00	2.00
Mistral Small 24B Instruct 250	4.00	4.00	6.00
Qwen3 235B A22B Instruct 2507	2.00	2.00	4.00
Google Gemma 3N E4B It	0.00	0.00	0.00

Table 2: Puzzle accuracy (%) by model and paradigm.

Results: Token Usage

	Single	SC	MAD
Prompt Tokens	281.4	1104.0	2209.6
Completion Tokens	250.0	1099.1	1081.2
Total Tokens	531.5	2203.1	3290.8
Prompt Ratio	1.00	3.92	7.85
Completion Ratio	1.00	4.40	4.32
Total Ratio	1.00	4.15	6.19

Table 5: Average token usage per paradigm.

Results: Inter-model Strength

Tournament Leaderboard

Rank	Player	Rating	Games
1	Stockfish	1871.5	70
2	gpt-3.5-turbo-instruct_SC_plan3	1735.1	71
3	gpt-3.5-turbo-instruct_SC	1726.3	72
4	gpt-3.5-turbo-instruct_plan3	1711.9	70
5	gpt-3.5-turbo-instruct	1678.3	71
6	mistralai/mistral-small-24b-instruct-2501	1132.3	70
7	deepseek-ai/deepseek-v3	1073.1	70
8	meta-llama/llama-3.3-70b-instruct	1071.5	70

Results: Deeper Planning Strength

- Multiple Plans help with
  - More stable chosen moves
  - Stronger play
- But planning is expensive
  - Higher token cost
  - Planning helps model win against stronger Stockfish levels
  - Used frequently
  - Planned moves succeed more often with fewer pieces on the board
- Solution
  - Enable planning only when number of pieces on the board falls below a threshold.

Setting	Plans	Stockfish depth	Length	% Planned	Tokens	Result
No Planning	0	5	294	0%	231041	0-1
Planning enabled	2 plans	5	71	41.7%	263411	1-0
Planning enabled	3 plans	7	99	26%	429019	1-0
Planning enabled	4 plans	7	83	33.3%	478143	1-0

Comparison of planning-enabled and no-planning games against Stockfish, in terms of total tokens used.

Results: Interpretability

- Factual grounding is very low
  - Reasoning rarely matches chosen moves
  - Grounding doesn’t correlate with correctness
- CoT might improve move selection, but not grounding[6]
- Models mostly perform sequence completion, not state-based reasoning.

Key Takeaways

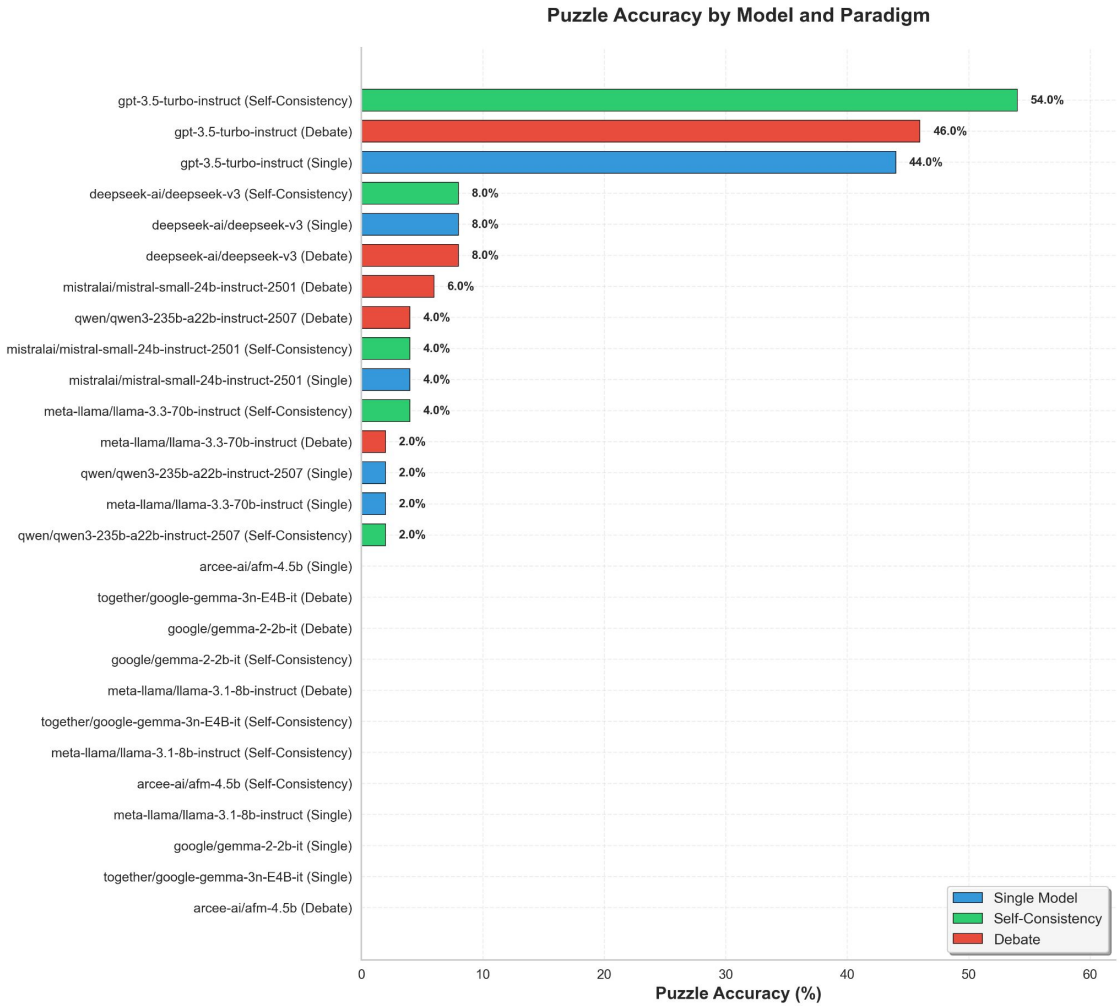
- Self-consistency, not debate, is the most reliable way to improve LLM reasoning in chess.
  - Debate adds large token cost with limited benefit.
- Open-source models consistently underperform closed models across all paradigms.
- Planning helps LLMs play stronger chess.
- Move quality does not correlate with reasoning quality
  - LLMs' plans and explanations are not faithful to their internal decision process.

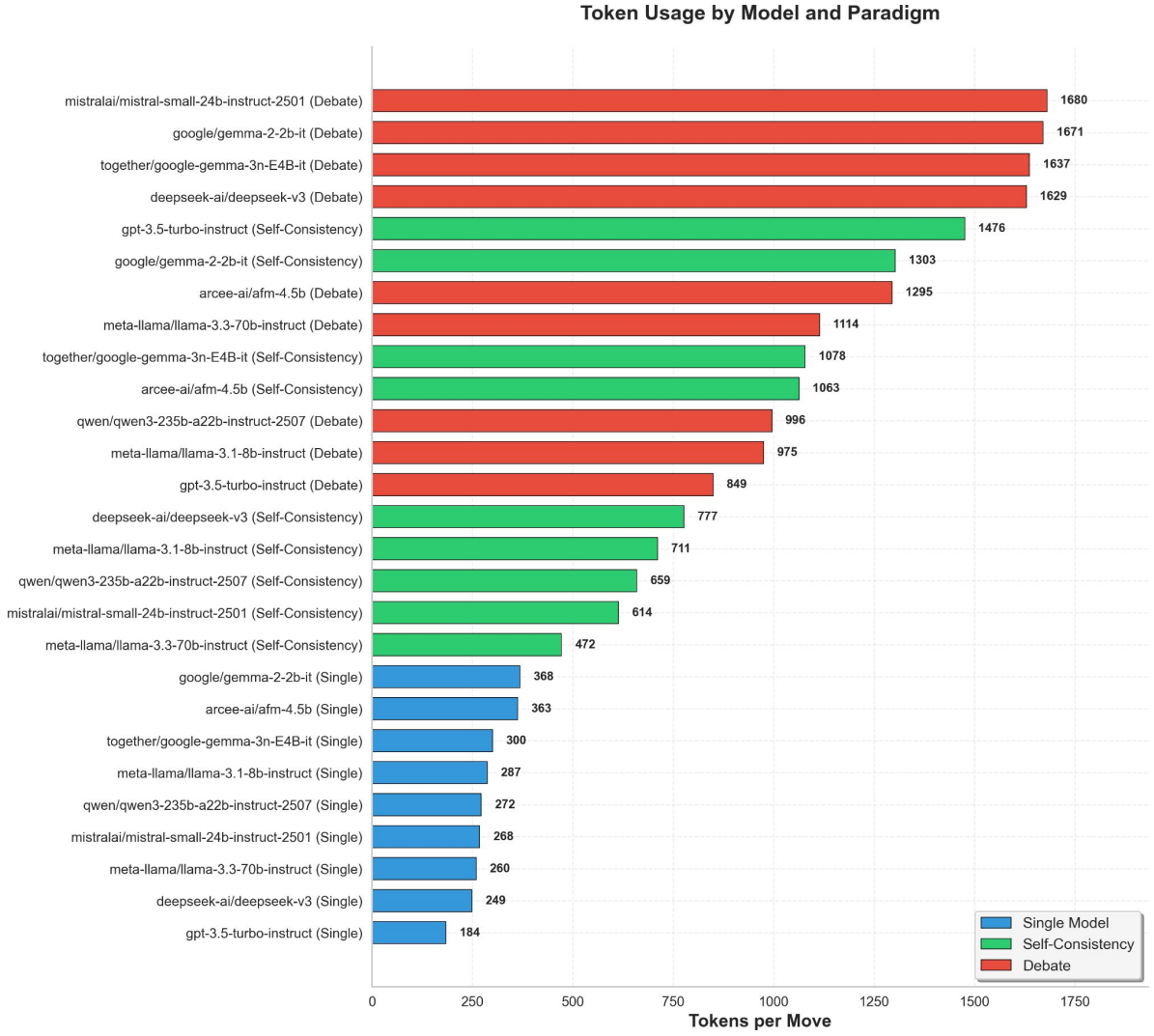
TODO final slides:

-

Puzzles

- GPT-3.5-Turbo-Instruct mysteriously good
- Open source models not performant




Token Usage by Model and Paradigm
mistralai/mistral-small-24b-instruct-2501 (Debate)
google/gemma-2-2b-it (Debate)
together/google-gemma-3n-E4B-it (Debate)
deepseek-ai/deepseek-v3 (Debate)
gpt-3.5-turbo-instruct (Self-Consistency)
google/gemma-2-2b-it (Self-Consistency)
arcee-ai/afm-4.5b (Debate)
meta-llama/llama-3.3-70b-instruct (Debate)
together/google-gemma-3n-E4B-it (Self-Consistency)
arcee-ai/afm-4.5b (Self-Consistency)
qwen/qwen3-235b-a22b-instruct-2507 (Debate)
meta-llama/llama-3.1-8b-instruct (Debate)
gpt-3.5-turbo-instruct (Debate)
deepseek-ai/deepseek-v3 (Self-Consistency)
meta-llama/llama-3.1-8b-instruct (Self-Consistency)
qwen/qwen3-235b-a22b-instruct-2507 (Self-Consistency)
mistralai/mistral-small-24b-instruct-2501 (Self-Consistency)
meta-llama/llama-3.3-70b-instruct (Self-Consistency)
google/gemma-2-2b-it (Single)
arcee-ai/afm-4.5b (Single)
together/google-gemma-3n-E4B-it (Single)
meta-llama/llama-3.1-8b-instruct (Single)
qwen/qwen3-235b-a22b-instruct-2507 (Single)
mistralai/mistral-small-24b-instruct-2501 (Single)
meta-llama/llama-3.3-70b-instruct (Single)
deepseek-ai/deepseek-v3 (Single)
gpt-3.5-turbo-instruct (Single)
1680
1671
1637
1629
1476
1303
1295
1114
1078
1063
996
975
849
777
711
659
614
472
368
363
300
287
272
268
260
249
184
Single Model
Self-Consistency
Debate
0
250
500
750
1000
1250
1500
1750
Tokens per Move

Errors by Model Size

Model Size	No Legal Move	Move Mismatch
<10B	415 (69.2%)	185 (30.8%)
10-50B	45 (31.5%)	98 (68.5%)
50-100B	90 (31.7%)	194 (68.3%)
100-200B	35 (44.9%)	43 (55.1%)
200B+	47 (32.2%)	99 (67.8%)

Planning

- Currently have implemented a way to prompt models to plan a few plies (half-moves) ahead.
- We capture this plan and then do not prompt the model again if the opponent played as expected
    - We expect this to be good for forced moves.
    - Qualitatively, it appears to save prompting for puzzles.
    - Models tend to generate extra moves anyways, so capturing the plan is helpful.

Wins!

- Multi-line planning helps!
- Each plan gives a different future continuation → more stable first moves
- Planning consistency improves as the number of plans increases

Full Game Results with Self-Consistency

- 2 plans per agent → wins against Stockfish at depth 5
- 4 plans per agent → wins against Stockfish at depth 7 (estimated elo of 2033)

Interpretability

- How accurately do models state factual claims about the board?
  - Frequently make incorrect statements about the board state
  - False claims do not necessarily lead to incorrect moves
  - Correct moves often come with incorrect reasoning

- LLMs don’t maintain an accurate board model during reasoning
- Next move selection relies on learned patterns

Remaining Plan

- Investigate quantitative tradeoffs for planning (e.g. token usage, game outcome improvements)
- Investigate inter-model game performance AKA **which models are better at chess against each other.**