Documentation: Selecting the Best LLM for Chess

1. Introduction

A key component of the project involves selecting the best large language model (LLM) capable of delivering optimal performance for the game. The selected models should not only play valid moves but also aim for a strategy close to specialized chess engines.

Our objective is to evaluate and compare non-specialized and fine-tuned LLMs to identify the most effective model. Also, identify and compare the performance of general-purpose LLMs and fine-tuned LLMs based on criteria such as **Elo rating**, move quality, and consistency with chess engines like Stockfish.

Stockfish is a powerful and widely used open-source chess engine. It is renowned for its incredible playing strength and has been a dominant force in computer chess competitions for years.

In our context, Stockfish acts as a reference point for benchmarking chess-playing LLMs. Since Stockfish is incredibly strong and consistent, comparing our model's performance (e.g., Elo rating, centipawn loss, or blunder rates) against Stockfish provides a reliable measure of the model's chess-playing ability.

2. Methodology

2.1. Benchmarking General LLMs

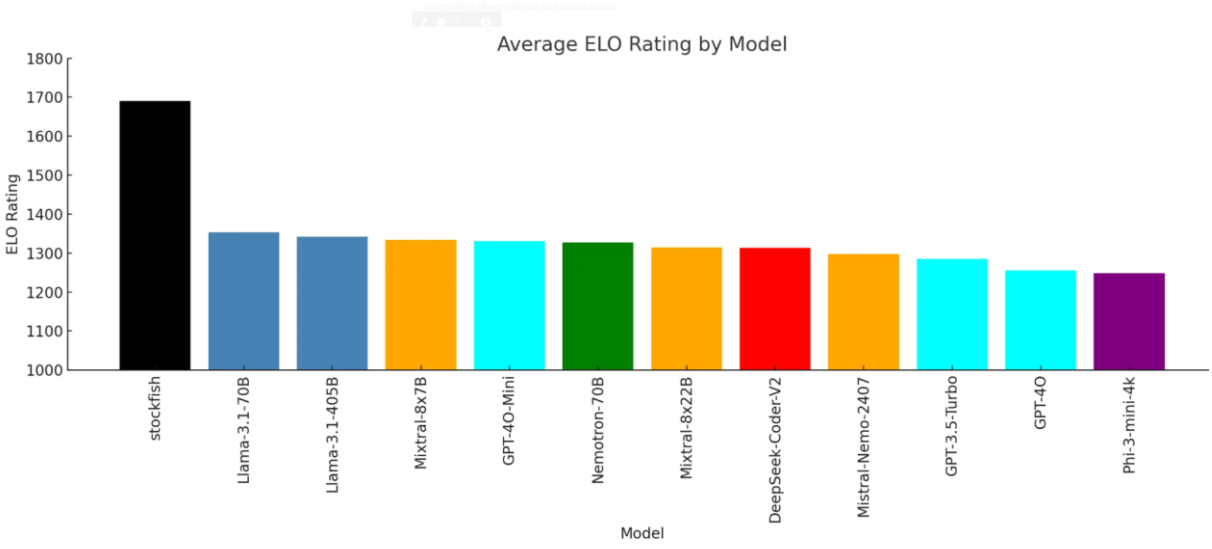
For the purpose of this benchmark, we decided to use Stockfish as the only reference. Also, instead of just relying on the outcome of the game (which is always a loss against the Stockfish engine anyway), we decided to use measures of the quality of moves produced during the game. These measures include:

- **The cumulative centipawn loss:** It measures how much a move deviates from the optimal or best possible move. The lower the better.
- **The blunder count:** This metric counts the number of moves that result in a significant drop in position value, typically defined as a centipawn loss of 100 or more. This metric helps identify severe mistakes during the game.
- **Inaccuracy count:** This metric counts the number of moves that result in moderate but notable positional losses, usually between 20 and 100 centipawns.
- **Matching moves top N:** The count of moves made by the model that match one of the top-N moves suggested by the engine. This metric helps in assessing how well the LLM can mimic high-level or optimal play.
- **Elo rating:** It is a numerical measure of the playing strength of the model, adjusted based on game performance and outcomes. In this case, the Elo rating is calculated for each game by supposing that the model and the Stockfish engine both have 1500 Elo rating at the start

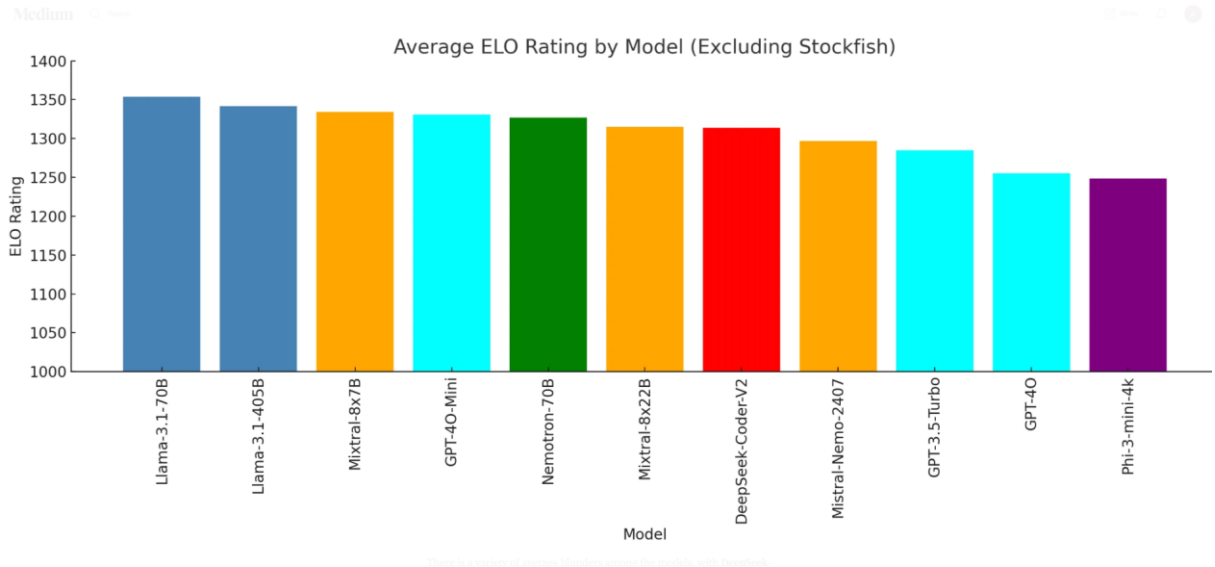
of each game. The idea is to evaluate the average rating loss of each model across several games against the Stockfish engine.

After running the experiment, we were able to draw some surprising and interesting observations and conclusions.

ELO ratings

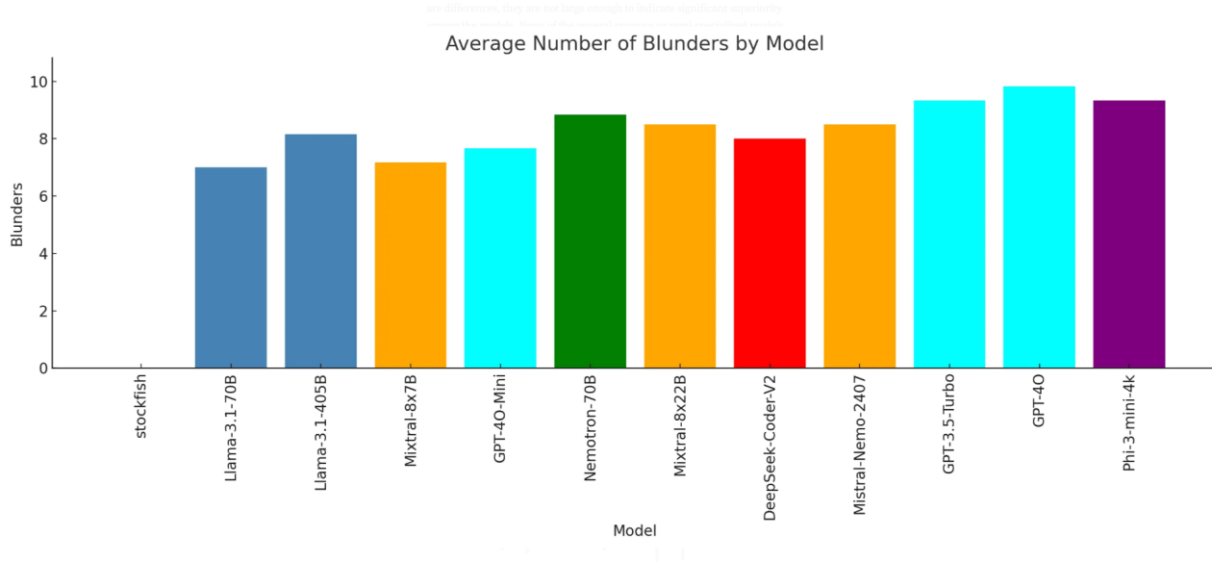


The average ELO ratings for various chess-playing models, excluding Stockfish, range from approximately **1248 to 1354**. The ELO rating is a measure of a model's playing strength, with higher values indicating stronger performance. The lowest ELO rating is observed in **Phi-3-mini-4k**, with an average of about 1248, while the highest is in **Llama-3.1-70B**, at around 1354. Models such as **Llama-3.1-70B** and **Nemotron-70B** are positioned at the upper end of the range, indicating relatively stronger play. Models like **Phi-3-mini-4k** and **GPT-4o** show lower ELO ratings, suggesting weaker relative performance in chess.



The narrow spread in ELO ratings (from 1248 to 1354) shows that while there are differences, they are not large enough to indicate significant superiority among the models. None of the general-purpose or semi-specialized models come close to challenging Stockfish's prowess. This suggests that while some models can play decently, their overall performance remains clustered in a moderate range, highlighting the limitations of non-specialized models.

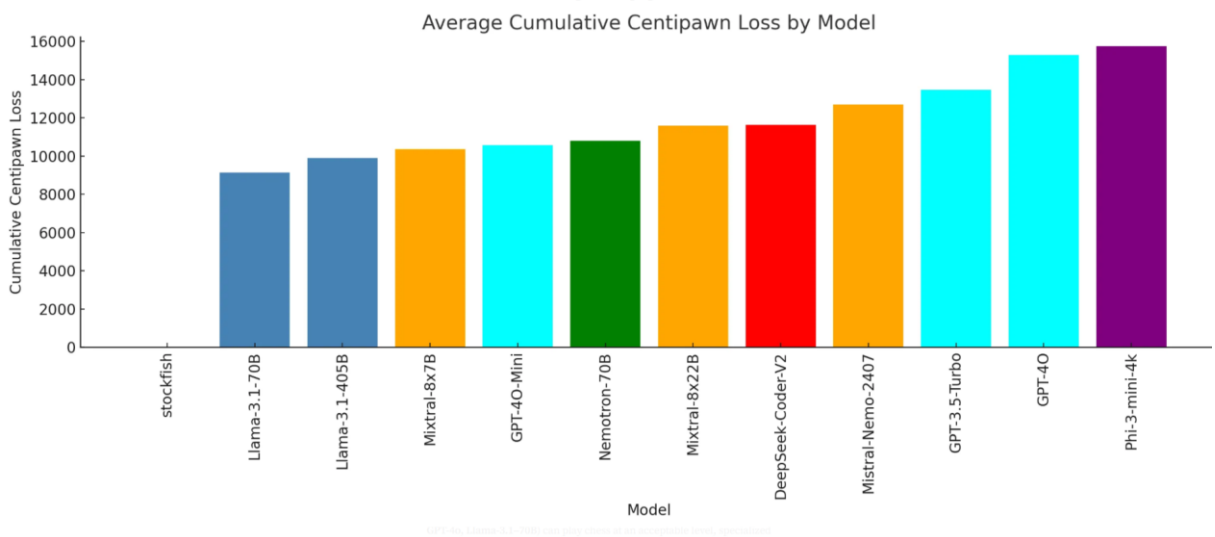
Blunders Analysis



There is a variety of average blunders among the models, with **DeepSeek-Coder-V2 and GPT-4o** showing the highest average counts, and models such as **Llama-3.1–70B** and **Mixtral-8x7B** showing fewer, which means they are more reliable.

The variation in blunders among models highlights differences in tactical soundness. Models with fewer blunders, such as **Llama-3.1-70B**, tend to perform better in practice, suggesting that lower blunder rates contribute to higher ELO ratings. Models that commit more blunders are less reliable in competitive scenarios.

Cumulative Centipawn Loss Analysis



The range for cumulative centipawn loss spans from lower values (indicating better precision) around **9000–11000** for models like **Llama-3.1-70B** to higher values exceeding **15000** for models like **Phi-3-mini-4k**.

Llama-3.1-70B, **Nemotron-70B**, and **Mixtral-8x22B** exhibit lower cumulative centipawn losses, suggesting more precise gameplay.

Phi-3-mini-4k and **DeepSeek-Coder-V2** have higher centipawn loss averages, indicating more frequent and significant deviations from optimal play.

Lower cumulative centipawn loss correlates with stronger, more consistent gameplay. The data shows that models like **Llama-3.1-70B** perform closer to optimal, whereas models with higher centipawn loss, such as **Phi-3-mini-4k**, make more suboptimal moves. **While no model matches Stockfish's zero centipawn loss, the spread indicates that certain general-purpose models are slightly better at maintaining closer-to-optimal moves than others.**

Overall, The analysis underscores that while general-purpose models (e.g., **GPT-4o, Llama-3.1-70B**) can play chess at an acceptable level, specialized models or those trained with chess-specific data (e.g., **Stockfish**) exhibit vastly superior performance.

Although the ELO and centipawn loss ranges between the models are not extreme, the absence of major outliers (other than Stockfish) points to general similarities in performance capabilities. This is coherent with the fact that these models have not been optimized for chess.

2.2. Benchmarking Fine-Tuned LLMs

Fine-Tuning Process

- **Dataset**:
  The models were fine-tuned using chess game datasets in PGN format, sourced from publicly available databases. These datasets include annotated game records, enabling the models to learn chess strategies, tactics, and positional understanding.
- **Methodology**:
  Each model was fine-tuned using frameworks like Hugging Face Transformers and PyTorch. The focus was on enhancing the models' chess-playing ability and evaluating their skill based on simulated games.
- **Objective**:
  The goal of fine-tuning was to enable the models to play chess at a level measurable by the **Elo Rating** system.

Evaluation and Results

- **Evaluation Metric**:
  Each model's performance was measured using **Elo Ratings**, calculated by playing games against Stockfish at a fixed intermediate level.
- **Results**:

Model	ELO Rating
mlabonne/chesspythia-70m	1202
EleutherAI/pythia-70m-deduped	1193
mlabonne/grandpythia-200k-70m	1061
facebook/opt-125m	1059
nlpguy/amdchess-v9	1058
bharathrajcl/chess_llama_68m	1056
Locutusque/TinyMistral-248M-v2.5	1055
nlpguy/amdchess-v2	1031
nlpguy/amdchess-v5	1027
Q-bert/ChessGPT	1026
Mxode/NanoLM-365M-Base	1024
nlpguy/amdchess	1023

Model	ELO Rating
amd/AMD-Llama-135m	1022
KoboldAI/fairseq-dense-125M	1022
EleutherAI/pythia-31m	1018
nlpguy/amdchess-v4	1017
lordspline/qwen-pruned-165m	1016
distilbert/distilgpt2	1016
EleutherAI/pythia-160m	1009
EleutherAI/pythia-14m	1009
nlpguy/smolchess-v2	1008
reflex-ai/AMD-Llama-350M-Upgraded	1007
Qwen/Qwen2.5-1.5B-Instruct	1005
EleutherAI/gpt-neo-125m	1004

Model	ELO Rating
nlpguy/smolchess	1002
ethzanalytics/pythia-31m	1000
stockfish	1000
EleutherAI/pythia-70m	1000
openai-community/gpt2-medium	998
UBC-NLP/Jasmine-350M	995
Qwen/Qwen2.5-3B	993
EleutherAI/pythia-2.8b	993
EleutherAI/pythia-160m-deduped	993
pgfeldman/GPT2-chess	993
chargoddard/SmolLlamix-8x101M	992
h2oai/h2o-danube2-1.8b-base	991

Model	ELO Rating
BlueSunflower/Pythia-160M-chess	990
superlazycoder/chesspythia-70m-random_1M	987
nlpguy/amdchess-v3	987
h2oai/h2o-danube3-500m-base	985
HuggingFaceTB/SmolLM2-135M	985
opencsg/csg-wukong-1B	982
TinyLlama/TinyLlama_v1.1	982
allenai/OLMo-1B-hf	979
pgfeldman/GPT2-Chess	975
Waterhorse/chessgpt-base-v1	973
raincandy-u/Quark-464M-v0.1.alpha	970
Qwen/Qwen2-0.5B	970

Model	ELO Rating
raincandy-u/Quark-464M-v0.1.alpha	970
Qwen/Qwen2-0.5B	970
AGundawar/chess-410m	965
Qwen/Qwen2.5-1.5B	958
appvoid/palmer-004	954
facebook/opt-350m	953
tiiuae/falcon-rw-1b	951
Qwen/Qwen2.5-0.5B	949
TinyLlama/TinyLlama-1.1B-Chat-v1.0	927
microsoft/Phi-3-mini-4k-instruct	922
mlabonne/chessgpt2-medium-smaller_pgn	919
microsoft/phi-2	851

Observations

- **mlabonne/chesspythia-70m** achieved the highest Elo rating of **1202**, demonstrating exceptional fine-tuning quality and dataset effectiveness.

- **EleutherAI/pythia-70m-deduped** followed closely with an Elo rating of **1193**, showcasing the potential of lightweight architectures when fine-tuned effectively.
- Models like **mlabonne/grandpythia-200k-70m** (Elo **1061**) and **facebook/opt-125m** (Elo **1059**) also performed well, likely benefiting from their specific architecture optimizations.
- The lower-performing models, such as **pgfeldman/GPT2-Chess** (Elo **975**) and **Waterhorse/chessgpt-base-v1** (Elo **973**), highlight the limitations in smaller architectures or less effective fine-tuning.
- The results suggest that model size, architecture, and fine-tuning quality are key factors in achieving high Elo ratings.

4. Discussion and Recommendations

4.1. Overall Comparison Between General and Fine-Tuned Models

- **Fine-Tuning Impact**:
  Fine-tuning significantly improves the Elo rating and overall chess-playing ability of the models. The top-performing fine-tuned models, such as **mlabonne/chesspythia-70m** and **EleutherAI/pythia-70m-deduped**, outperformed general-purpose LLMs by a substantial margin.
- **General Models' Limitations**:
  General-purpose models like **facebook/opt-125m** and **microsoft/phi-2** lag behind fine-tuned counterparts due to their lack of domain-specific training. However, they serve as robust baselines for further specialization.

4.2. Recommended Model

- Based on the benchmarking results, **mlabonne/chesspythia-70m** emerges as the recommended model for this project. It achieved the highest Elo rating (**1202**) among all evaluated models, demonstrating exceptional performance in chess-related tasks.
- For applications requiring a trade-off between computational efficiency and performance, **EleutherAI/pythia-70m-deduped** is a viable alternative with an Elo rating of **1193**.

4.3. Areas for Improvement

- **Dataset Diversity**:
  To further improve fine-tuned models, additional training on larger and more diverse chess datasets is recommended. Including games from players of varying skill levels and unique scenarios could enhance model adaptability.
- **Advanced Techniques**:
  Investigating reinforcement learning methods, such as **AlphaZero-like frameworks**, could complement fine-tuning and push the models' performance closer to human expertise.

- **Model Scaling**:
  Exploring larger-scale models or ensemble approaches may yield improvements, albeit at a higher computational cost.

5. Conclusion

The comparison highlights the transformative impact of fine-tuning in adapting general-purpose LLMs for specialized tasks like chess. **mlabonne/chesspythia-70m** proved to be the most effective, achieving an Elo rating of **1202** and setting a benchmark for further development.