Transforming Checkmate: Direct Preference Optimization and Self-Play

Darrell Best
dsbest@usc.edu

Jacob Lichtefeld
jal_690@usc.edu

Nicklas Holmberg
nholmber@usc.edu

Loring Scotty Hoag
lhoag@usc.edu

Robert Pabo
pabo@usc.edu

Shravya Shashidhar
sshashid@usc.edu

Abstract

Recent work has demonstrated that transformers can achieve grandmaster-level chess play (2895 ELO) without explicit search through large-scale supervised training on billions of positions. We investigate whether Direct Preference Optimization (DPO) can further enhance strategic reasoning in already-strong models through post-training on engine-derived preferences. Using the Searchless Chess 9M-parameter model, we apply DPO with Stockfish-ranked move preferences. While preference accuracy improves dramatically from 12% to 89%, actual playing performance collapses: win rates drop from 50% to 9% against the baseline and puzzle-solving accuracy falls from 87% to 67%. Analysis reveals a fundamental incompatibility between DPO's pairwise training objective and greedy action selection. DPO provides training signals for only two moves per position, leaving remaining legal moves unconstrained. These "intruder" moves drift to high q-values and dominate action selection despite not being targeted by optimization. Our negative results highlight important challenges in applying preference-based alignment methods to complex reasoning domains with discrete action spaces.

1 Introduction

Reasoning remains one of the most fundamental challenges in artificial intelligence. Although LLMs have achieved remarkable success in natural language tasks, their ability to perform structured and logical reasoning remains questionable. Chess provides a nearly ideal testbed for investigating this limitation: It demands precise calculation, strategic planning over long horizons, and the ability to generalize across an astronomical number of possible board configurations.

Traditional chess engines have relied on explicit search algorithms, from minimax with alpha-beta pruning in classical engines like Stockfish (The Stockfish developers, 2025) to neural network-guided MCTS in AlphaZero (Silver et al., 2018). These approaches fundamentally depend on exploring future game states. In contrast, transformer models operate through pattern recognition and learned representation. The question of whether transformers can develop genuine reasoning capabilities for chess without explicit search has been partially answered by recent work: Ruoss et al. (2024b) demonstrated that transformers trained on 10 billion positions can achieve grandmaster-level play (2895 ELO) without any search, suggesting that transformers can indeed internalize sophisticated strategic reasoning at sufficient scale.

However, this breakthrough raises a new question: Can post-training techniques further enhance the strategic reasoning of already-strong searchless chess models? While supervised learning on billions of positions proves effective, self-play has driven some of the most impressive achievements in game-playing AI, from AlphaGo to AlphaZero. Moreover, modern language model alignment techniques like Direct Preference Optimization (Rafailov et al., 2023) offer stable, sample-efficient alternatives to traditional reinforcement learning. Our work investigates whether combining self-play with DPO can improve transformer-based chess models beyond what supervised learning alone achieves, potentially revealing new pathways for developing AI systems capable of systematic logical thought.

2 Related Work

LLMs for Chess. Recent investigation into LLMs for chess have revealed fundamental limitations in strategic reasoning. Noever et al. (2023) trained GPT-2 models on PGN chess games, achieving intermediate level play at approximately 1500 ELO. While their Chess Transformer could learn legal moves and basic strategies, performance plateaued

far below expert level, suggesting inherent architectural constraints in transformer-based approaches to chess. Li et al. (2024) conducted a comprehensive evaluation of state-of-the-art LLMs including GPT-3.5 and GPT-4 on chess reasoning tasks. Their analysis revealed severe limitations: models produced illegal moves 20-50% of the time and consistently failed at multi-step tactical calculations. Even when generating legal moves, the models struggled with position evaluation and could not maintain consistent game state representations, highlighting the gap between pattern matching and genuine chess understanding.

Search-Augmented Methods. Feng et al. (2023) took a different approach by combining MCTS with transformer-based policy networks. Their ChessGPT system achieved strong performance exceeding 2000 ELO, but critically, this success depended on explicit search algorithms rather than pure language modeling. This work demonstrates that while transformers can learn effective policy representations, they require search augmentation to achieve competitive play. Ruoss et al. (2024a) explored whether transformers could internalize search through amortized planning. By training models to predict action values derived from Stockfish evaluations, they achieved up to 2000 ELO on familiar positions without explicit search. However, performance degraded significantly on novel positions outside the training distribution, suggesting that transformers struggle to generalize strategic reasoning beyond memorized patterns.

Searchless Chess Models. Most relevant to our work, Ruoss et al. (2024b) achieved a breakthrough by demonstrating that transformers trained on 10 billion positions can reach grandmaster-level play (2895 ELO) without tree search at inference time. Their models use distributional reinforcement learning with action-value representations, outputting probability distributions over returns for each legal move. This work provides our foundation: we investigate whether post-training with DPO can further improve these already-strong models.

Preference Optimization. Direct Preference Optimization (Rafailov et al., 2023) has emerged as an effective alternative to traditional RLHF for language model alignment. DPO directly optimizes models using preference pairs without requiring a separate reward model, achieving strong results in natural language tasks. However, its application to complex reasoning domains like chess with action-value models remains underexplored.

Parameter-Efficient Fine-Tuning. Given the computational demands of training chess-specific models, parameter-efficient methods offer practical advantages. Hu et al. (2022) introduced Low-Rank Adaptation (LoRA), enabling efficient fine-tuning through low-rank weight decompositions. Zhang et al. (2023) extended this with adaptive parameter allocation, while Dettmers et al. (2023) combined quantization with LoRA to further reduce memory requirements.

3 Building Our Own Searchless Chess Model

Before the DPO optimization final approach, we also explored fine-tuning our own searchless transformer-based language model to generate legal chess moves in PGN/FEN notation. While this effort ran into practical limitations based on computation time requirements, we report our experiments in this space which will show initial success at fine-tuning open-source language models to play chess.

3.1 Fine Tuning GPT-OSS

In an effort to reproduce similar work fine-tuning a GPT model on Forsyth–Edwards Notation (FEN), we set up the capability to train GPT-OSS with LoRA (Hu et al., 2022) so that rather than optimizing all 20 billion parameters we could tune a smaller number closer to previous work. The hope was that the larger model would be more promising even if we were not training the full weights. We successfully implemented a training pipeline limited to single GPU usage on ISI's Eclair Cluster.[1]

Figure 1 shows the performance of one of our fine-tuned GPT-OSS models at various stages of training. Each training step uses a total batch size of 128 and takes approximately 10 seconds on an NVIDIA A6000 GPU. The training demonstrates steady improvement in puzzle accuracy while reducing illegal move rates, validating our implementation. However, contemporary work suggests achieving grandmaster-level ELO would require training on 20,000,000+ examples, which would take a single A6000 over 6 months of continuous computation.

Given the time constraints of this project, we made a strategic decision to pivot to using pre-

[1]We spent considerable effort attempting to get multi-GPU training working on the compute environment, but a variety of technical issues prevented deployment within our timeline.

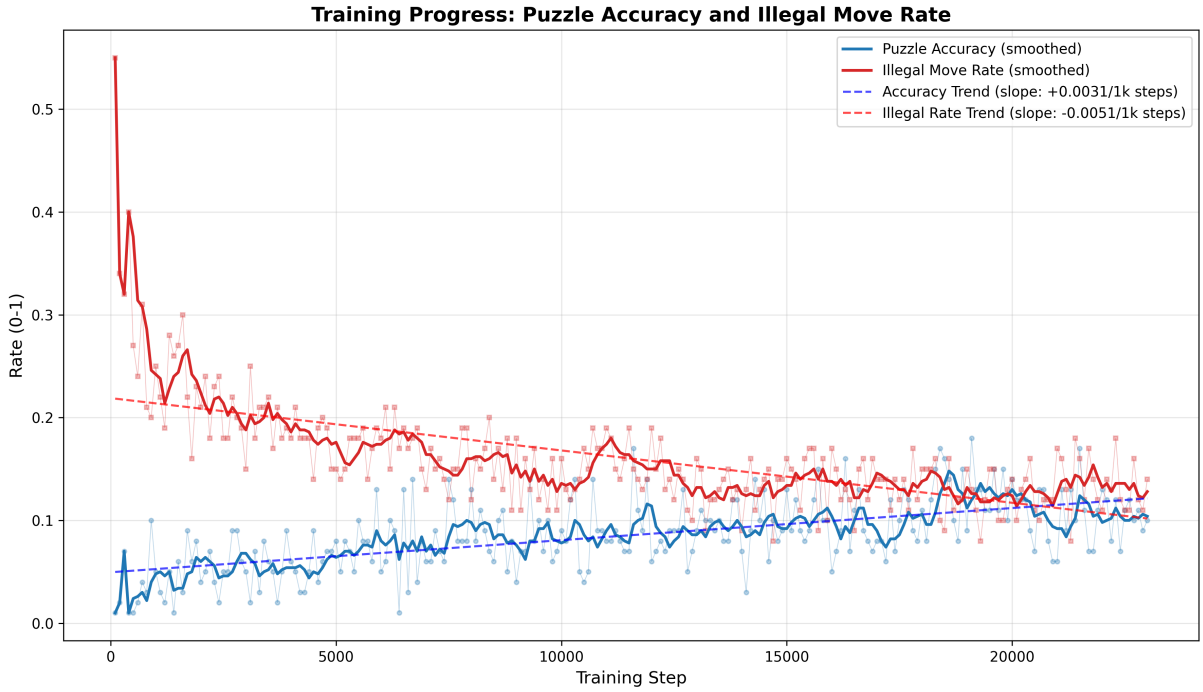

Figure 1: Performance of fine-tuning GPT-OSS with LoRA, showing steady improvement. Red line is the illegal move rate and blue line is puzzle performance.

trained models from Ruoss et al. (2024b). Their Searchless Chess models (9M, 136M, and 270M parameters) are publicly available and already achieve strong performance, with the 270M model reaching 2895 ELO. This pivot allows us to focus our limited time on our primary research question: whether self-play with Direct Preference Optimization can further improve already-strong chess models, rather than spending months replicating baseline training from scratch.

4 Improving Searchless Chess via DPO

Can modern post-training techniques improve already-strong searchless chess models? Building upon the Searchless Chess models, we investigate whether self-play with Direct Preference Optimization can further enhance strategic reasoning through the following approach: (i) Present the model with chessboard positions from Lichess (ii) Use Stockfish's pre-computed best moves to identify positions where the model made suboptimal moves. (iii) Create preference pairs (Stockfish's move as "chosen" vs. model's move as "rejected"). (iv) Apply DPO to train the model to prefer better moves. This approach combines the exploration benefits of self-play (similar to AlphaZero (Silver et al., 2018)) with the stability and sample efficiency of DPO (Rafailov et al., 2023), avoiding the complexity of traditional reinforcement learning's reward modeling and value function bootstrapping.

4.1 Datasets and Models

Base Models: We utilize the pretrained Searchless Chess models (Ruoss et al., 2024b). These models come in three sizes: (i) 9M parameters (baseline: 1800 ELO) (ii) 136M parameters (baseline: 2600 ELO) (iii) 270M parameters (baseline: 2895 ELO)

All models were trained on 10 billion positions generated by Stockfish, using distributional reinforcement learning with action-value representations. They output probability distributions over returns for each legal move in a given position, encoding both policy and value information in a single model.

Training Data Generation: For our DPO training, we use the Lichess evaluation dataset, which contains chess positions annotated with Stockfish's top move evaluations. We generated approximately 1 million DPO preference pairs by comparing the base model's predicted moves against Stockfish's recommendations. When the model's top move did not appear in Stockfish's top-ranked moves for a position, we created a preference pair with Stockfish's best move as "chosen" and the model's move as "rejected."

Evaluation Data: We evaluate model improvements using:

- **Lichess Puzzle Database**: 100-1,000 tactical puzzles across difficulty ratings (1000-2500+ ELO) for measuring puzzle-solving accuracy
- **Head-to-head tournaments**: Models play 50-200 games against their base versions to compute ELO ratings via established chess rating formulas
- **Stockfish analysis**: Centipawn loss (raync910), a metric measuring positional evaluation differences, on held-out game positions to quantify move quality

All datasets are publicly available or generated programmatically, requiring no manual data collection.

4.2 DPO Setup

We focused our DPO experiments on the 9M parameter Searchless Chess model to enable faster iteration cycles, with plans to scale to larger models pending positive results. The 9M model provides a reasonable baseline at approximately 1800 ELO while requiring significantly less compute than the 136M or 270M variants.

Hyperparameters: We explored learning rates ranging from 1×10^{-3} to 3×10^{-4}, with the KL divergence coefficient β varied between 0.5 and 5.0 to control the strength of the preference optimization relative to the reference policy. Lower β values allow more aggressive updates toward preferred moves, while higher values keep the model closer to its original behavior.

Training Infrastructure: We used NVIDIA DGX Spark systems for rapid prototyping and hyperparameter exploration, then scaled to ISI's Eclair cluster for longer training runs. This two-tier approach allowed us to quickly identify promising configurations before committing to extended training.

5 Results

We evaluate our DPO-trained model across three key metrics: preference accuracy, head-to-head tournament play, and puzzle-solving ability. All experiments use the 9M parameter Searchless Chess model as the baseline.

5.1 Preference Accuracy

Table 1 shows that DPO successfully optimizes the intended objective. The model's preference accuracy (defined as the fraction of positions where the model assigns a higher q-value to Stockfish's preferred move over the model's original choice) improves dramatically from 12% to 89% after DPO training. This confirms that the DPO loss function operates as expected and the model learns to prefer engine-approved moves in the pairwise comparison setting.

Model	Preference Accuracy
Baseline (9M)	12%
DPO-trained	89%

Table 1: Preference accuracy on held-out test positions. Preference accuracy measures whether the model assigns higher q-value to Stockfish's preferred move compared to the model's original choice.

5.2 Tournament Performance

Despite strong preference accuracy, actual playing performance degrades substantially. Table 2 shows results from 100 games between the DPO-trained model and the baseline. The DPO model achieves only a 9% win rate, compared to the expected 50% win rate of equally-matched opponents. This represents a catastrophic performance collapse despite optimizing for Stockfish-approved moves.

Model	Wins	Losses	Win Rate
Baseline (9M)	91	9	91%
DPO-trained	9	91	9%

Table 2: Head-to-head tournament results over 100 games. The DPO-trained model performs dramatically worse than the baseline despite improving preference accuracy.

5.3 Puzzle Solving

Table 3 shows performance on 1,000 tactical puzzles from the Lichess database. The baseline 9M model achieves 87% accuracy, while the DPO-trained model drops to 67% accuracy. This 20 percentage point decrease indicates that DPO training degrades the model's tactical calculation abilities despite optimizing for better moves in isolation.

5.4 Analysis of Failure Mode

Our results reveal a fundamental mismatch between DPO's training objective and the model's inference mechanism. While DPO successfully teaches the model to prefer Stockfish's moves over the model's original mistakes in pairwise comparisons (89%

Model	Puzzle Accuracy
Baseline (9M)	87%
DPO-trained	67%

Table 3: Puzzle-solving accuracy on 1,000 Lichess tactical puzzles spanning difficulty ratings from 1000-2500 ELO.

preference accuracy), this does not translate to improved move selection during actual play.

Investigation of the q-value distributions reveals the core issue: while the preferred move's q-value increases relative to the rejected move, other candidate moves (which we term "intruders") maintain or develop higher q-values than our target move. Since the Searchless Chess models use greedy action selection (argmax over all legal moves), these intruder moves are selected at inference time despite our training signal. The model learns the correct pairwise ordering between chosen and rejected moves, but this local improvement fails to produce global move selection improvement across all 30+ legal moves in typical positions.

6 Discussion

6.1 The Intruder Problem

The intruder problem stems from a fundamental mismatch between DPO's pairwise training objective and the model's greedy inference mechanism. Standard DPO optimizes for relative preferences between exactly two actions: the chosen move and the rejected move. However, chess positions typically have 30 or more legal moves, and DPO provides no training signal for the vast majority of these alternatives.

During training, while we successfully push the chosen move's q-value above the rejected move's, the remaining legal moves (the "intruders") are unconstrained. These moves can drift to arbitrarily high q-values without penalty, and since the model uses greedy selection (argmax over all legal moves), an intruder with a higher q-value than our target move will be selected at inference time.

This represents a structural limitation: DPO was designed for language model alignment where outputs are sampled according to probability distributions, not selected via argmax over action values. The Searchless Chess architecture's greedy selection mechanism is fundamentally incompatible with DPO's pairwise optimization approach.

6.2 Why Preference Accuracy Increased While Performance Collapsed

Our results demonstrate a critical disconnect between optimization objectives and downstream performance. DPO successfully achieves its stated goal (teaching the model to prefer better moves in pairwise comparisons) yet this produces catastrophic performance degradation. This paradox reveals several insights:

Local vs. Global Optimization. DPO optimizes local pairwise preferences without ensuring global consistency across all legal moves. In language modeling, where outputs are sampled from probability distributions, small inconsistencies are averaged out. In chess, where greedy selection chooses the single highest-valued move, any inconsistency in the ranking produces incorrect actions.

Q-Value Distribution Collapse. We observed that q-values across all moves became increasingly pessimistic during training, with the overall magnitude of predicted returns decreasing. This suggests that DPO's contrastive objective may destabilize the carefully-calibrated q-value distributions learned during the original supervised training on billions of positions.

Training-Inference Mismatch. The Searchless Chess models were trained with distributional RL objectives that explicitly model the full return distribution for each action. DPO, in contrast, only provides pairwise comparison signals. This mismatch may explain why our training signal fails to produce meaningful improvements despite optimizing the intended pairwise objective.

6.3 Attempted Mitigation

We attempted to address the intruder problem by dynamically generating DPO pairs for intruder moves as they emerged during training. When an intruder's q-value exceeded the target move's, we created a new preference pair with the original target as chosen and the intruder as rejected. However, this approach proved ineffective: while q-values shifted slightly, the overall ordering of moves remained unchanged. The model appeared to reach an equilibrium where our training signal could not overcome the underlying q-value structure.

6.4 Implications for Future Work

Our negative results highlight several important considerations for applying preference optimization to complex reasoning domains:

Architecture Compatibility. Preference-based methods like DPO may be fundamentally incompatible with greedy action-selection architectures. Future work should either: (i) modify the architecture to use sampling-based action selection compatible with preference learning, or (ii) develop preference optimization methods that explicitly handle multi-way rankings across all actions rather than pairwise comparisons.

Alternative Post-Training Approaches. Rather than DPO, other post-training techniques may prove more effective. Possibilities include: (i) continued supervised learning on higher-quality game data, (ii) behavior cloning from stronger Searchless Chess variants (training the 9M model to imitate the 270M model), or (iii) hybrid approaches that combine retrieval-augmented generation for openings and endgames with the core learned policy.

Evaluation Metrics. Our work underscores the importance of measuring downstream task performance rather than proxy metrics. High preference accuracy proved to be a misleading indicator of actual playing strength, suggesting that alignment research should prioritize end-to-end evaluation over intermediate objectives.

7 Additional Exploration Approaches

7.1 RAG Infrastructure for Potential Extension

In parallel with our main efforts, we developed retrieval-augmented generation (RAG) infrastructure that could potentially augment the DPO-trained models if time permits after completing the primary experiments. This system uses FAISS for efficient similarity search over chess positions.

Implementation: We built a RAG system with custom embeddings that combine board state representations with Stockfish position evaluations. Position-only embeddings proved insufficient since structurally different positions can have similar strategic implications. The database is populated with opening theory from established chess databases and endgame tablebases for positions with ≤ 7 pieces.

8 Conclusion

This work investigates whether Direct Preference Optimization can improve already-strong searchless chess models through post-training on engine-derived preferences. Our results reveal a fundamental incompatibility between DPO's pairwise training objective and the greedy action-selection mechanism used by the Searchless Chess models.

While DPO successfully optimizes its stated objective (preference accuracy improves from 12% to 89%), this optimization produces catastrophic downstream performance degradation. Win rates against the baseline collapse from 50% to 9%, and puzzle-solving accuracy drops from 87% to 67%. This disconnect arises because DPO provides training signals for only two moves per position, leaving the remaining 30+ legal moves unconstrained. These "intruder" moves drift to high q-values during training and dominate greedy action selection at inference time, despite not being targeted by the optimization objective.

Our findings have several important implications. First, preference-based alignment methods developed for language modeling may not transfer effectively to domains with discrete action selection. Language models sample from probability distributions, naturally averaging over small ranking inconsistencies, while chess models use argmax selection where any ranking error produces incorrect behavior. Second, proxy metrics like preference accuracy can be highly misleading indicators of downstream performance. Third, the carefully-calibrated q-value distributions learned during supervised pre-training appear fragile under post-training optimization, with DPO's contrastive objective causing distribution collapse.

Our earlier experiments with fine-tuning GPT-OSS using LoRA demonstrated that parameter-efficient training can produce legal and moderately competent chess play. However, scaling to grandmaster-level performance requires computational resources (6+ months of continuous GPU training) beyond the scope of this project, motivating our pivot to investigating post-training improvements of existing strong models.

Future work should explore: (i) multi-way ranking methods that provide training signals for all legal moves rather than pairwise comparisons, (ii) sampling-based action selection architectures compatible with preference learning, (iii) behavior cloning from stronger model variants, and (iv) hybrid retrieval-augmented approaches combining learned policies with perfect play databases for openings and endgames. Our negative results provide valuable insights for the broader challenge of applying language model alignment techniques to complex reasoning domains.

References

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36:10088–10115.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Mguni, Yali Du, and Jun Wang. 2023. Chessgpt: Bridging policy learning and language modeling. *Advances in Neural Information Processing Systems*, 36:7216–7262.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Xiang Li, Yiming Zhang, and Dong Yang. 2024. Beyond next-move prediction: Evaluating large language models on chess reasoning and legality. *arXiv preprint arXiv:2401.04521*.

David Noever, Matt Ciolino, and Josh Kalin. 2023. The chess transformer: Mastering play using generative language models. *arXiv preprint arXiv:2303.08383*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36.

raync910. Average centipawn loss in chess (acpl). Chess.com Blog. Accessed: 2024.

Anian Ruoss, Grégoire Delétang, Sourabh Medapati, Jordi Grau-Moya, Li K Wenliang, Elliot Catt, John Reid, Cannada A Lewis, Joel Veness, and Tim Genewein. 2024a. Amortized planning with large-scale transformers: A case study on chess. *Advances in Neural Information Processing Systems*, 37:65765–65790.

Anian Ruoss, Grégoire Delétang, Sourabh Medapati, Jordi Grau-Moya, Li Kevin Wenliang, Elliot Catt, John Reid, and Tim Genewein. 2024b. Grandmaster-level chess without search. *arXiv preprint arXiv:2402.04494*. NeurIPS 2024.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, and 1 others. 2018. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144.

The Stockfish developers. 2025. Stockfish. Repository: `https://github.com/official-stockfish/Stockfish`.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. Adalora: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*.