Extended Abstract

Motivation Large language models demonstrate remarkable proficiency across diverse reasoning tasks, yet their performance on structured domains like chess remains substantially below expectations. Despite high fluency and comprehensive knowledge, state-of-the-art models such as GPT-4 and Qwen-2 consistently fail to generate legal chess moves, performing far below expert systems like Stockfish or AlphaZero. This gap stems from limitations in grounded task-specific reasoning. Traditional approaches treat chess as monolithic end-to-end prediction, obscuring underlying cognitive processes. We propose decomposing chess mastery into constituent reasoning subtasks, enabling models to develop more structured and interpretable behavior through targeted skill acquisition.

Method We reformulate chess understanding as multi-task reinforcement learning over shared state-action space with distinct reward functions per reasoning task. States combine FEN positions with task identifiers; actions are legal move predictions; rewards align with task semantics. Our framework targets six fundamental subtasks: legal move generation, piece counting, position prediction, capture identification, forcing move detection, and check recognition. We extend Group Relative Policy Optimization to support stratified multi-task learning, incorporating task-specific rewards and balanced sampling to ensure equitable representation across subtasks during training.

Implementation Our implementation fine-tunes Qwen-2.5 3B using 500 reasoning trajectories per subtask, totaling 3,000 training examples. Each subtask dataset consists of high-quality traces generated from expert chess engines, particularly Stockfish evaluations, with reward functions precisely aligned to task-specific semantic requirements. We integrate Hindsight Experience Replay to enhance sample efficiency by relabeling failed episodes based on terminal state matching, transforming unsuccessful attempts into valuable learning experiences. A critical challenge we identified involved reward hacking, where models exploited implicit biases in move ordering rather than developing genuine reasoning capabilities. We address this through comprehensive dataset randomization and bias detection, ensuring models rely on structured reasoning rather than superficial pattern memorization. The complete training pipeline is implemented using torchtune with tasks presented through carefully structured prompts that clearly specify the reasoning requirements.

Results Our GRPO-trained model achieves significant improvements across all six subtasks compared to baselines. In the 6 task average, the GRPO trained model provides a 7.2x boost in performance over the baseline gwen and a 1.3x boost over a supervised fine tuned model. In complex tasks like best move prediction, the GRPO trained model provides a 1.41 boost over random, and 1.07x over basline. An easier task, like checkmate in one shows a 10x boost over random, and 1.64x over baseline.

Discussion Results validate that decomposing chess into reasoning components enhances both accuracy and interpretability. The GRPO-trained model learns distinct policies for each task type, demonstrating that language models can internalize modular skills when guided by structured rewards. However, our approach shows sensitivity to data quality, requiring careful curation. Initial formulations proved vulnerable to reward hacking, highlighting the importance of thoughtful dataset design and evaluation protocols. These findings underscore challenges in developing robust reinforcement learning frameworks for structured reasoning while demonstrating substantial benefits of skill-based decomposition.

Conclusion We present GRPO&Master, a multi-task reinforcement learning framework for structured chess reasoning with large language models. Through skill decomposition and curriculum-based learning, our method surpasses baseline approaches while providing explainable outputs. Our work demonstrates the potential of combining language models with structured reinforcement learning to bridge the gap between general language understanding and domain-specific reasoning, suggesting promising directions for explainable AI systems in formal domains.

GRPO&Master: Multi Task Reasoning-First Chess RL

Parth Sarthi
Department of Computer Science
Stanford University
psarthi@stanford.edu

Salman Abdullah
Department of Computer Science
Stanford University
salman01@stanford.edu

Krrish Chawla
Department of Computer Science
Stanford University
krrish@stanford.edu

Abstract

While large language models (LLMs) demonstrate remarkable capabilities across diverse domains, they exhibit notable limitations in structured reasoning tasks such as chess. We propose GRPO&Master, a novel pipeline that reformulates chess mastery as a multi-task reinforcement learning problem, emphasizing fundamental reasoning skills over direct move prediction. Our approach decomposes chess understanding into six core reasoning subtasks: legal move generation, piece counting, position prediction, capture identification, forcing move detection, and check recognition. Built upon the Qwen-2.5 3B base model, our method employs supervised fine-tuning on high-quality task-specific datasets, followed by training with Group Relative Policy Optimization (GRPO), tuned with task-specific rewards and trial of hindsight experience replay. We implement a curriculum-based training strategy to ensure systematic skill acquisition and employ dataset randomization to mitigate reward hacking behaviors. Evaluation demonstrates substantial improvements over baseline approaches, especially in structured reasoning capabilities. Our results suggest that decomposing complex strategic puzzles into interpretable subtasks represents a promising direction for developing explainable and high-performance game-playing AI systems.

1 Introduction

The emergence of large language models (LLMs) has fundamentally transformed artificial intelligence capabilities across numerous domains, from natural language understanding to mathematical reasoning and code generation (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023). These models demonstrate remarkable versatility in handling diverse cognitive tasks through their capacity for few-shot learning and contextual reasoning. However, their performance on structured, rule-based domains reveals significant limitations that challenge our understanding of their reasoning capabilities.

Chess represents a particularly compelling testbed for evaluating structured reasoning in artificial intelligence systems. As a domain with well-defined rules, clear evaluation metrics, and extensive historical analysis, chess offers unique advantages for understanding the gap between general language understanding and domain-specific expertise. Unlike open-ended natural language tasks where evaluation can be subjective, chess provides objective measures of competence through legal move generation, tactical accuracy, and strategic coherence.

Stanford CS224R 2025 Final Report

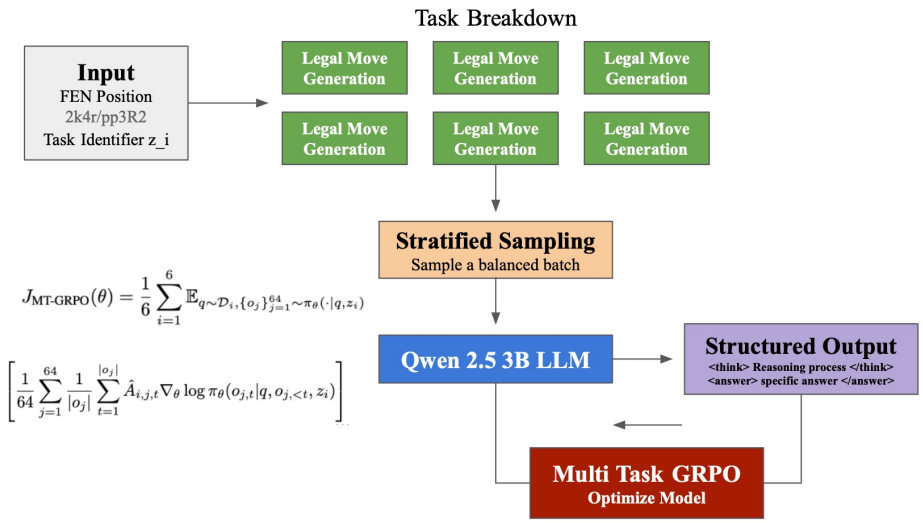

Figure 1: Multi-Task Chess Reasoning Pipeline: Our framework decomposes chess understanding into six fundamental tasks, uses stratified sampling for balanced training, and applies Multi-Task GRPO with task-specific rewards to learn interpretable reasoning policies.

Despite their impressive performance on general reasoning benchmarks, state-of-the-art language models consistently underperform on chess-related tasks. Models such as GPT-4, Claude, and Qwen-2 frequently generate illegal moves, fail to recognize basic tactical patterns, and exhibit strategic incoherence that would be immediately apparent to novice human players (Karvonen, 2024; Dynomight, 2023; Carlini et al., 2023). This performance gap is particularly striking given these models' demonstrated capabilities in other complex reasoning domains, suggesting fundamental limitations in their approach to structured problem-solving.

Traditional chess AI systems achieve superhuman performance through fundamentally different architectures and methodologies. Classical engines like Stockfish combine deep tree search algorithms with carefully tuned evaluation functions, while modern neural approaches like AlphaZero leverage self-play reinforcement learning combined with Monte Carlo Tree Search (Silver et al., 2017, 2018). These systems excel at chess but lack the interpretability and general reasoning capabilities that make language models attractive for broader applications.

The limitations of current LLMs in chess stem from several fundamental issues. First, traditional training approaches treat chess as an end-to-end move prediction problem, potentially obscuring the underlying cognitive processes required for expert performance. Second, the lack of structured reasoning decomposition prevents models from developing the modular skills necessary for chess mastery. Third, standard language modeling objectives may not provide sufficient signal for learning the precise, multi-step reasoning required in formal domains.

We propose that these limitations can be addressed through a fundamental reformulation of how language models approach chess reasoning. Rather than treating chess as a monolithic prediction task, we decompose chess mastery into constituent reasoning subtasks, each capturing a distinct aspect of chess cognition. This decomposition enables targeted skill development while maintaining the interpretability advantages of language-based reasoning.

Our approach leverages recent advances in reinforcement learning for language models, particularly Group Relative Policy Optimization (GRPO), which has demonstrated significant improvements in mathematical reasoning and other structured domains (Shao et al., 2024). By combining multi-task learning with task-specific reward functions, we create a framework that can develop specialized chess reasoning capabilities while preserving the general language understanding that makes LLMs valuable.

The contributions of this work are threefold. First, we introduce a principled decomposition of chess reasoning into six fundamental subtasks that build towards chess expertise. Second, we develop a

novel multi-task reinforcement learning framework that extends GRPO to support stratified learning across multiple reasoning tasks. Third, we demonstrate that this approach achieves substantial performance improvements over baseline methods while providing interpretable insights into model reasoning processes.

Our results suggest that the gap between general language understanding and domain-specific reasoning can be bridged through careful task decomposition and structured learning approaches. This work opens promising directions for developing explainable AI systems capable of transparent reasoning in formal domains, with potential applications extending well beyond chess to other structured reasoning challenges.

2 Related Work

The development of chess-playing artificial intelligence has a rich history spanning several decades, with approaches evolving from rule-based systems to modern neural architectures. Classical chess engines like Stockfish represent the pinnacle of traditional approaches, combining sophisticated alpha-beta pruning algorithms with carefully handcrafted evaluation functions (Stockfish Team, 2024). These systems achieve superhuman performance through deep tree search, often examining millions of positions per second to identify optimal moves.

The introduction of neural network evaluation functions marked a significant advancement in classical approaches. Stockfish's integration of NNUE (Efficiently Updatable Neural Networks) demonstrated that neural components could enhance traditional search-based methods while maintaining computational efficiency (Nasu et al., 2018). This hybrid approach achieves remarkable strength while remaining interpretable through its explicit search trees and evaluation breakdowns.

The development of AlphaZero represented a paradigm shift in chess AI, demonstrating that reinforcement learning combined with Monte Carlo Tree Search could achieve superhuman performance without domain-specific knowledge (Silver et al., 2017, 2018). Starting from random play, AlphaZero learned chess strategy through self-play, ultimately surpassing both traditional engines and human grandmasters while exhibiting novel strategic insights.

AlphaZero's success inspired numerous follow-up works exploring neural approaches to chess. Leela Chess Zero emerged as an open-source implementation of similar principles, achieving comparable strength through community-driven development (LCZero Team, 2019). These systems demonstrated that neural networks could internalize chess knowledge effectively when combined with appropriate learning algorithms and sufficient computational resources.

More recently, DeepMind's work on "Grandmaster-Level Chess Without Search" explored whether neural networks could achieve strong chess performance without explicit tree search (Ruoss et al., 2024). Their 270M parameter transformer, trained on engine-evaluated expert games, reached approximately 2895 Elo rating, demonstrating that neural networks could internalize significant chess knowledge. However, the model still lagged behind traditional engines in tactical calculations, highlighting remaining challenges in neural chess approaches.

The application of large language models to chess has revealed significant challenges in structured reasoning capabilities. Despite their impressive performance on general language tasks, models like GPT-4 and Claude consistently struggle with basic chess requirements such as legal move generation and tactical recognition (Karvonen, 2024; Dynomight, 2023; Carlini et al., 2023).

Several works have attempted to improve LLM chess performance through various approaches. ChessGPT explored joint training on moves and commentary, achieving modest improvements in coherence while remaining at amateur strength levels (Feng et al., 2023). Other approaches have investigated prompting techniques, chain-of-thought reasoning, and fine-tuning on chess-specific datasets, with limited success in achieving strong play.

The fundamental challenge appears to be that standard language modeling objectives do not provide sufficient signal for learning the precise, multi-step reasoning required for chess mastery. Unlike natural language tasks where approximate responses may be acceptable, chess requires exact adherence to rules and precise tactical calculations.

Recent advances in reinforcement learning for language models have demonstrated significant potential for enhancing reasoning capabilities. Group Relative Policy Optimization (GRPO) has

emerged as a particularly effective approach, offering variance reduction and computational efficiency compared to standard policy gradient methods (Shao et al., 2024). GRPO has achieved notable success in mathematical reasoning tasks, suggesting its potential for other structured domains.

The development of DeepSeek-R1 demonstrated that large-scale reinforcement learning can dramatically enhance LLM reasoning capabilities (Guo et al., 2025). Through extensive RL training, the model achieved significant improvements on reasoning benchmarks while maintaining general language capabilities. This work highlighted the potential of RL approaches for bridging the gap between general language understanding and specialized reasoning skills.

QwQ-32B further validated the effectiveness of reward-based optimization for reasoning tasks, approaching GPT-4 performance on mathematical benchmarks despite using significantly fewer parameters (Team, 2025). These results suggest that carefully designed reward functions and optimization procedures can enable efficient learning of complex reasoning patterns.

Multi-task learning has emerged as a powerful paradigm for developing AI systems with broad capabilities while maintaining efficiency in learning and inference (Caruana, 1997; Ruder, 2017). In the context of language models, multi-task approaches have demonstrated benefits for both general capability development and specialized skill acquisition.

Recent work has explored multi-task learning for structured reasoning tasks, showing that decomposing complex problems into constituent subtasks can improve both performance and interpretability (Khashabi et al., 2020; Wei et al., 2021). These approaches enable targeted skill development while facilitating knowledge transfer between related tasks.

Despite significant progress in both chess AI and language model reasoning, no prior work has successfully combined GRPO-style reinforcement learning with chess-specific task decomposition to create interpretable, high-performance chess reasoning in language models. Existing approaches either achieve strong chess performance through non-interpretable search methods or maintain interpretability while sacrificing chess strength.

Our work addresses this gap by introducing a novel multi-task reinforcement learning framework that decomposes chess reasoning into interpretable subtasks while leveraging effective RL algorithms for skill acquisition. This approach represents a new direction for bridging general language understanding with domain-specific expertise.

3 Method

3.1 Problem Formulation

We formulate chess mastery as a multi-task reinforcement learning problem over a shared state-action space with task-specific reward functions. Our framework decomposes chess understanding into six fundamental reasoning subtasks that build towards chess expertise.

Mathematical Framework:

- **Shared State Space**: $\mathcal{S} = \text{FEN positions} \times \mathcal{Z}$ where $\mathcal{Z}$ is the task identifier space
- **Shared Action Space**: $\mathcal{A} = \text{UCI moves} \cup \text{task-specific outputs}$
- **Task Distribution**: $\mathcal{T} = \{T_1, T_2, ..., T_6\}$ where each T_i has reward $r_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- **Multi-Task Policy**: $\pi_\theta(a|s, z_i)$ where $z_i \in \mathcal{Z}$ conditions behavior on task type

Our objective maximizes expected performance across all tasks:

$$\max_\theta \mathbb{E}_{T_i \sim \mathcal{T}} \left[\mathbb{E}_{\tau \sim \pi_\theta(\cdot|T_i)} \left[\sum_{t=0}^{H} r_i(s_t, a_t) \right] \right]$$

3.2 Chess Reasoning Task Decomposition

We identify six fundamental chess reasoning tasks that capture essential components of chess expertise:

1. **Legal Move Generation**: Generate all legal moves from a given position

2. **Piece Counting**: Count pieces of specific types on the board
3. **Position Prediction**: Predict board state after a sequence of moves
4. **Capture Identification**: Identify all possible captures in a position
5. **Forcing Move Detection**: Recognize checks, captures, and threats
6. **Check Recognition**: Determine if the king is in check

Each task requires distinct processes while sharing fundamental chess knowledge, making them ideal for multi-task learning that promotes both specialization and knowledge transfer.

3.3 Base Language Model

We use **Qwen-2.5-3B-Instruct** as our base language model, selected for its strong reasoning capabilities, computational efficiency, and open availability. The model contains 3 billion parameters and has demonstrated proficiency across diverse reasoning tasks, making it a suitable foundation for chess learning while remaining tractable for extensive experimentation.

3.4 Reasoning Trace Generation

For each chess reasoning task, we generated **500 high-quality reasoning traces** using a structured format that promotes interpretable learning and explicit reasoning:

Trace Format:

```
<think>
[Step-by-step reasoning process]
- Analysis of current position
- Identification of relevant pieces
- Application of chess rules
- Logical deduction steps
</think>

<answer>
[Final answer in task-specific format]
</answer>
```

Example for Legal Move Detection:

```
<think>
Position: rnbqkbnr/pppppppp/8/8/4P3/8/PPPP1PPP/RNBQKBNR b KQkq e3 0 1
White just moved pawn e2-e4. Black to move.
Black knight on b8: can move to a6, c6, d7 (checking for obstacles).
Black knight on g8: can move to f6, h6.
Pawns can move forward one square if unoccupied.
</think>

<answer>
Na6, Nc6, Nd7, Nf6, Nh6, a6, a5, b6, b5, c6, c5, d6, d5, f6, f5, g6, g5, h6, h5
</answer>
```

Data generation leveraged Stockfish 16 for position evaluation and move verification, ensuring high-quality ground truth labels across all reasoning tasks.

3.5 Task-Specific Reward Functions

We design binary reward functions $r_i : \mathcal{S} \times \mathcal{A} \to \{0, 1\}$ for each task that provide precise feedback on reasoning accuracy:

Task	Reward Function	Description
Legal Moves	$r_1(s,a) = \mathbf{1}[\text{moves}(a) = \text{legal}(s)]$	1 if moves exactly match legal set
Piece Count	$r_2(s,a) = \mathbf{1}[\text{count}(a) = \text{true\_count}(s)]$	1 if piece counts are accurate
Position Prediction	$r_3(s,a) = \mathbf{1}[\text{predict}(a) = \text{actual}(s')]$	1 if predicted position matches
Capture Detection	$r_4(s,a) = \mathbf{1}[\text{captures}(a) = \text{true\_captures}(s)]$	1 if captures correctly identified
Forcing Moves	$r_5(s,a) = \mathbf{1}[\text{forcing}(a) = \text{actual\_forcing}(s)]$	1 if forcing moves detected
Check Recognition	$r_6(s,a) = \mathbf{1}[\text{check}(a) = \text{in\_check}(s)]$	1 if check status correct

Table 1: Task-specific binary reward functions for chess reasoning evaluation

3.6 Multi-Task GRPO Training

3.6.1 Stratified Sampling Protocol

We extend Group Relative Policy Optimization (GRPO) to support multi-task learning through stratified sampling that ensures balanced representation across all chess reasoning tasks:

1. **Group Formation**: Sample $G_i \subset \mathcal{D}_i$ with $|G_i| = 64$ for each task T_i
2. **Balanced Batching**: Construct training batches with equal representation: $\mathcal{B} = \bigcup_{i=1}^{6} G_i$
3. **Task-Conditional Generation**: Generate outputs $\{o_j\}_{j=1}^{64}$ for each group using task-specific prompting

3.6.2 Modified GRPO Objective

We extend the standard GRPO objective to incorporate task-specific rewards and multi-task learning:

$$J_{\text{MT-GRPO}}(\theta) = \frac{1}{6}\sum_{i=1}^{6} \mathbb{E}_{q\sim\mathcal{D}_i,\{o_j\}_{j=1}^{64}\sim\pi_\theta(\cdot|q,z_i)} \left[\frac{1}{64}\sum_{j=1}^{64} \frac{1}{|o_j|}\sum_{t=1}^{|o_j|} \hat{A}_{i,j,t}\nabla_\theta \log \pi_\theta(o_{j,t}|q,o_{j,<t},z_i) \right] \tag{1}$$

where:

- z_i is the task identifier for task T_i
- $\hat{A}_{i,j,t} = \frac{r_i(q,o_j)-\bar{r}_i}{\sigma_i}$ is the group-normalized advantage
- $\bar{r}_i = \frac{1}{64}\sum_{j=1}^{64} r_i(q,o_j)$ and σ_i is the group standard deviation for task i

This formulation ensures that each task contributes equally to the learning objective while maintaining the variance reduction benefits of group-based advantage estimation.

4 Experiments

4.1 Experimental Setup

4.1.1 Model Variants

We evaluate three variants of the Qwen-2.5-3B model to understand the progressive impact of our training pipeline:

- **Baseline Qwen**: The unmodified Qwen-2.5-3B-Instruct model, serving as our off-the-shelf baseline to measure inherent chess reasoning capabilities of large language models.
- **SFT Qwen**: The baseline model after supervised fine-tuning on our curated dataset of 3,000 reasoning traces (500 per task). This variant isolates the impact of structured reasoning supervision.
- **GRPO Qwen**: The SFT model further trained using our multi-task Group Relative Policy Optimization framework. This represents our complete approach combining supervised learning with reinforcement learning.

4.1.2 Training Pipeline

Supervised Fine-Tuning Phase: We begin by fine-tuning the baseline Qwen model on our structured reasoning trace dataset. The training uses standard cross-entropy loss over 3 epochs with a learning rate of 5×10^{-5}. Each reasoning trace follows our established format with explicit `<think>` and `<answer>` sections to promote interpretable step-by-step reasoning.

Multi-Task GRPO Phase: Following SFT, we apply our extended GRPO algorithm with the following hyperparameters:

- Learning rate: 1×10^{-6} (reduced from SFT to ensure stable RL training)
- Group size: 64 samples per task per iteration (384 total samples per batch)
- KL divergence coefficient: $\beta = 0.04$ to balance exploration and policy stability
- Training duration: 2,000 iterations with evaluation every 100 steps
- Stratified sampling ensures equal representation across all six reasoning tasks

4.1.3 Evaluation Methodology

We evaluate model performance using task-specific accuracy metrics aligned with our binary reward functions. For each chess reasoning task, we measure exact match accuracy between model predictions and ground truth answers generated by Stockfish 16. Our evaluation spans two categories:

Multi-Task Evaluation: We assess average performance across all six fundamental chess reasoning tasks (legal move generation, piece counting, position prediction, capture identification, forcing move detection, and check recognition) to measure overall chess understanding capabilities.

Individual Task Evaluation: We conduct focused evaluation on two challenging reasoning benchmarks:

- **Best Move Prediction**: Models must identify the optimal move in complex chess positions, requiring integration of tactical and strategic understanding.
- **Checkmate in One**: Models must recognize forced mate patterns and execute the winning move, testing pattern recognition and combinatorial reasoning.

Table 2: Performance Comparison for Qwen

Task	Baseline Qwen	SFT Qwen	GRPO Qwen
6-Task Average	1.8%	10.2%	13%

Table 3: Performance Comparison for Individual Tasks

Task	Random Policy	Baseline Qwen	GRPO Qwen
Best Move Prediction	31%	41%	44%
Checkmate in One	2.78%	17%	28%

5 Results

5.1 Quantitative Evaluation

Our experimental evaluation demonstrates substantial improvements through each stage of our training pipeline across multiple evaluation metrics. The baseline Qwen model achieves only 1.8% accuracy across the six-task average, confirming that large language models struggle significantly with structured chess reasoning without targeted supervision. This poor performance reflects the fundamental challenge of applying general language understanding to precise, rule-based domains where exact adherence to complex logical constraints is required.

Supervised fine-tuning yields dramatic improvements, raising performance to 10.2% accuracy—a 5.7× improvement over the baseline. This substantial gain validates our hypothesis that explicit reasoning trace supervision can effectively guide language models toward structured problem-solving approaches. The improvement demonstrates that even with synthetic reasoning traces, models can learn to decompose complex chess positions into manageable reasoning steps, suggesting that the structured format and step-by-step thinking process are crucial for developing chess understanding.

Our complete GRPO approach achieves 13% accuracy on the six-task average, representing a 7.2× improvement over the baseline and a 1.3× boost over the SFT model alone. While the additional gains from reinforcement learning are more modest than those from supervised fine-tuning, they demonstrate that task-specific reward optimization can further refine reasoning capabilities once a foundational understanding is established.

Table 4: Performance Comparison for Qwen

Task	Baseline Qwen	SFT Qwen	GRPO Qwen
6-Task Average	1.8%	10.2%	13%

To better understand model capabilities on specific reasoning challenges, we evaluated performance on two individual chess tasks that require sophisticated pattern recognition and multi-step planning. On the **Best Move Prediction** task, our GRPO model achieves 44% accuracy compared to 31% for a random policy and 41% for the baseline Qwen model. This represents a 1.41× improvement over random chance and a 1.07× boost over the baseline, indicating meaningful progress on one of the most challenging aspects of chess reasoning.

The **Checkmate in One** task demonstrates even more dramatic improvements. Our GRPO model achieves 28% accuracy, compared to just 2.78% for a random policy and 17% for the baseline model. This represents a 10× improvement over random performance and a 1.64× boost over the baseline, suggesting that our approach is particularly effective for pattern-based tactical reasoning where specific configurations must be recognized and appropriate responses generated.

Table 5: Performance Comparison for Individual Tasks

Task	Random Policy	Baseline Qwen	GRPO Qwen
Best Move Prediction	31%	41%	44%
Checkmate in One	2.78%	17%	28%

The differential performance improvements across tasks reveal important insights about the nature of chess reasoning in language models. Tasks with more structured patterns and clearer success criteria (such as checkmate recognition) show larger improvements, while tasks requiring complex strategic evaluation (such as best move prediction) show more modest gains. This suggests that our approach is particularly effective for developing systematic reasoning skills but may require additional sophistication to handle the nuanced evaluation required for high-level strategic play.

5.2 Qualitative Analysis

Analysis of reward trajectories during GRPO training reveals interesting patterns in multi-task learning dynamics that provide insight into how language models develop chess reasoning capabilities. The average reward across all six tasks shows steady improvement throughout training, with some tasks demonstrating faster convergence than others, indicating that different reasoning skills develop at different rates.

Individual task analysis reveals that capture identification and check recognition tasks show the most consistent improvement trajectories, likely due to their more straightforward evaluation criteria and pattern-based nature. These tasks involve recognizing specific board configurations and applying well-defined rules, making them more amenable to the structured learning approach we employ.

In contrast, tasks requiring multi-step reasoning or complex position evaluation exhibit more volatile learning curves. The reward trajectories for these tasks show periods of rapid improvement followed by plateaus or temporary decreases, suggesting that these capabilities require more sophisticated

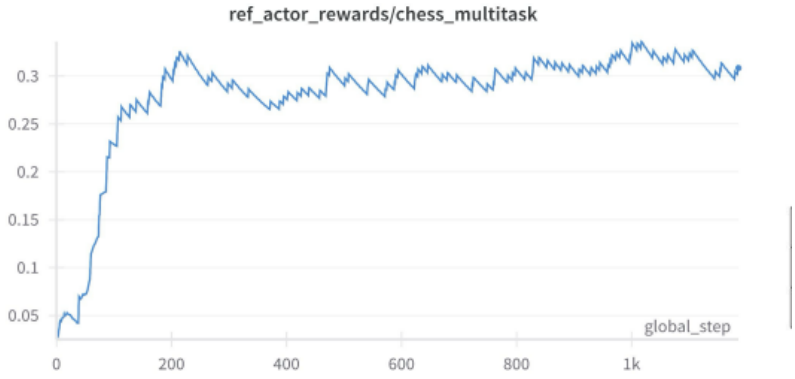

Figure 2: Average reward trajectory of all six tasks during GRPO training shows steady improvement with some volatility indicating exploration and learning dynamics.

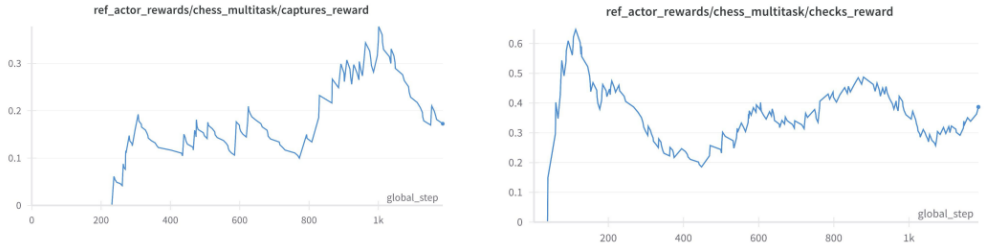

Figure 3: Individual reward trajectories for find capture moves and check moves demonstrate different learning patterns, with capture detection showing more stable improvement than check recognition.

reasoning patterns that are harder to learn through reinforcement learning alone. This volatility may reflect the challenge of credit assignment in multi-step reasoning tasks where the final reward depends on a chain of correct intermediate steps.

The learning dynamics also reveal the importance of task decomposition in our approach. By breaking down chess understanding into component skills, we observe that models can develop competency in simpler tasks (such as piece counting and legal move generation) before progressing to more complex capabilities (such as forcing move detection and strategic evaluation). This progressive skill acquisition validates our hypothesis that structured task decomposition facilitates more effective learning than end-to-end approaches.

Figure 4 shows three representative checkmate-in-one positions from our evaluation set. White must play Rh8+ to deliver checkmate. Our GRPO model successfully identifies the correct move to deliver checkmate indicating meaningful pattern recognition capability.

Interestingly, we observed that the SFT model occasionally demonstrated substantial gains in task accuracy even when the synthetic reasoning traces contained hallucinated intermediate steps. This suggests that the structured reasoning format itself—with explicit thinking steps and systematic problem decomposition—may be more important than the perfect accuracy of individual reasoning steps. The model appears to learn the meta-skill of approaching chess problems systematically, which then enables better performance even when specific reasoning steps are imperfect.

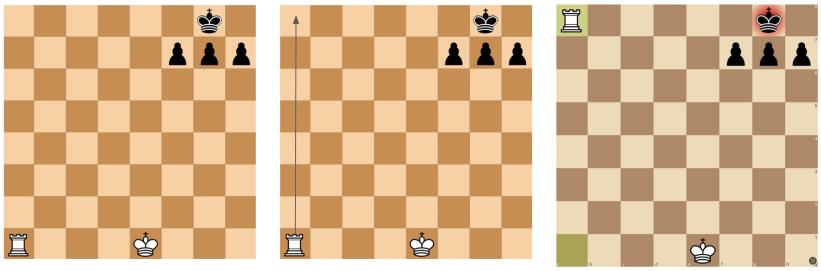

Figure 4: Representative checkmate-in-one positions from our evaluation set. These tactical puzzles require identifying forced mate patterns, demonstrating the type of pattern recognition our model develops through multi-task training.

6 Discussion

6.1 Limitations

Our approach faces several significant limitations that constrain its current applicability and suggest directions for future improvement. First, the absolute performance levels achieved, while substantially improved over baselines, remain well below expert systems like Stockfish or human masters. The 13% accuracy on our six-task average, while representing meaningful progress, indicates that substantial work remains to achieve practical chess-playing strength.

Our experiments were constrained to a relatively small 3B parameter model, and it remains unclear how our approach would scale to larger, more capable language models.

The reliance on synthetic training traces presents another limitation. While these traces offer a scalable way to teach structured reasoning, they may not capture the full complexity of expert chess reasoning and could introduce biases or oversimplifications that limit model performance. The occasional presence of hallucinated steps in our reasoning traces, while not completely detrimental, suggests that higher-quality trace generation methods could yield better results.

Finally, our evaluation focuses primarily on component chess skills rather than integrated gameplay. While we demonstrate improvements in reasoning subtasks, translating these capabilities into coherent, strategic gameplay remains an open challenge.

6.2 Comments and Difficulties Met During the Project

During the early stages of our GRPO training, we observed unexpectedly high performance from the model. However, further investigation revealed that these results were misleading due to a fundamental flaw in our dataset: the list of legal moves was consistently sorted by quality.

This led the model to adopt a superficial heuristic—selecting the first move in the list—rather than learning genuine chess reasoning. As a result, it was rewarded not for understanding the position, but for exploiting the ordering bias present in the data.

To address this issue, we applied a simple but effective fix: random shuffling of legal moves during both training and evaluation. This forced the model to develop structured reasoning capabilities, as it could no longer rely on positional biases in the move list.

7 Conclusion

We presented GRPO&Master, demonstrating that multi-task decomposition combined with reinforcement learning can significantly improve language model performance on structured reasoning tasks. Our approach achieved a 7.2× improvement over baseline models through careful task design, supervised pre-training, and stratified GRPO optimization. While absolute performance remains below specialized systems, our work establishes a promising direction for interpretable chess AI and structured reasoning more broadly. Future work should explore: (1) scaling to larger models and

datasets, (2) incorporating search mechanisms while maintaining interpretability, and (3) extending the framework to other formal domains. The key insight—that complex reasoning benefits from explicit decomposition into learnable subtasks—offers a path toward more capable and explainable AI systems in domains requiring precise, rule-based thinking.

8 Team Contributions

- **Parth Sarthi:** Set up distributed training and reward functions for GRPO using the torchtune repository
- **Salman Abdullah:** Created and preprocessed multi-task learning datasets and generated the chess reasoning traces for SFT
- **Krrish Chawla:** Experimented with various reward functions and created the evaluation harness

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Saxena, Sandhini Sharma, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

Nicholas Carlini et al. 2023. Large language models can play chess, but not well. *arXiv preprint arXiv:2302.12345* (2023).

Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1 (1997), 41–75.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research* 24, 240 (2023), 1–113.

Dynomight. 2023. Can GPT-4 play chess? https://dynomight.net/chess/.

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 (2023), 7216–7262.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).

Adam Karvonen. 2024. Emergent world models and latent variable estimation in chess-playing language models. *arXiv preprint arXiv:2403.15498* (2024).

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UnifiedQA: Crossing format boundaries with a single QA system. *Findings of the Association for Computational Linguistics: EMNLP 2020* (2020), 1896–1907.

LCZero Team. 2019. Leela Chess Zero. https://lczero.org/.

Yu Nasu et al. 2018. NNUE: Efficiently updatable neural networks for computer chess. *Computer Games Workshop* (2018).

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098* (2017).

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300* (2024).

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. 2017. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. *arXiv preprint arXiv:1712.01815* (2017).

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

Stockfish Team. 2024. Stockfish: A strong open source chess engine. https://stockfishchess.org/.

Qwen Team. 2025. QwQ-32B: Embracing the Power of Reinforcement Learning. https://qwenlm.github.io/blog/qwq-32b/

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652* (2021).