Even LLMs deserve a body: How grounded chess representations impact language-mediated strategic reasoning

Anonymous
Anonymous Affiliation
email@anonymous.com

Abstract

While Large Language Models excel in many cognitive-demanding domains such as mathematics and programming, they still significantly struggle in others, such as playing chess. Seeking to overcome this limitation, in this paper we propose a novel multimodal architecture which projects the embeddings of a neural network trained to play chess into the latent space of a language model. We demonstrate that our approach vastly outperforms traditional text-based fine-tuning, obtaining up to 34 percentage points higher puzzle accuracy. Crucially, our evidence suggests the model does not simply relay the best move from the chess encoder. Rather, it actively combines the grounded board representation with linguistic reasoning to achieve these gains. Finally, as ongoing work, we propose a pipeline to isolate whether these gains stem from the latent encoding of the board features, or if providing the same information in natural language would yield equivalent results.

1 Introduction

"The ability to play chess is the sign of a gentleman. The ability to play chess well is the sign of a wasted life." allegedly asserted the first chess world champion, Paul Morphy. Based on recent findings by Kolasani *et al*. [2025] even frontier Large Language Models can be largely considered gentlemen. Despite solving increasingly complex problems in cognitive-demanding domains such as programming and mathematics [Kwa *et al.*, 2026; Verbeken *et al.*, 2026], their chess skills have not improved at the same pace, with many models unable to win against an opponent making random moves [Kolasani *et al.*, 2025]. Hwang *et al*. [2025], inline with recent Reinforcement Learning literature [Zhao *et al.*, 2025; Li *et al.*, 2025], suggest that even chess-focused fine-tuning cannot overcome these limitations, which stem from an insufficient exposure to chess-related content during pre-training.

First, we make the following preliminary observations:

- McGrath *et al*. [2022] show that AlphaZero, a neural network which attains super-human playing strength, independently *rediscovers* most human chess concepts, which can be linearly decoded from its embeddings, suggesting that its latent space shares some common features with the one developed by human chess experts.
- Secondly, Jhamtani *et al*. [2018] reports that chess corpora features highly pragmatic language, omitting many nuanced board features that appear trivial to an experienced chess player, an issue known in the NLP literature as *reporting bias* [Gordon and Van Durme, 2013].
- Lastly, Kamath *et al*. [2026] have recently argued that vision-language models trained on data affected by high reporting bias do not develop emerging reasoning abilities, no matter the scale of the model or the dataset. Instead, a focused fine-tuning that incorporates the implicit, *tacit* reasoning traces omitted in the original dataset allows to overcome these limitations.

For these reasons, we propose a novel approach to enhance the strategic reasoning of Language Models in the game of chess: we project into a language model a new *chess* modality, grounded in the embeddings of an expert neural network. We believe that integrating an expert system that has developed a latent chess space comparable to that of human experts, such as LeelaChessZero[1], offers a principled solution to the reporting bias problem. Rather than crafting an extensive dataset that verbalizes all the tacit chess reasoning patterns, we only need a compact alignment dataset to teach the language model to map Leela's latent space back into language. This way, the LLM can simply leverage the grounded, strategic knowledge already encoded within the chess embeddings rather than re-learning it from scratch.

We present empirical evidence that this approach improves the model's strategic reasoning and chess-playing strength, crucially showing that the model does not simply relay the best move predicted by the engine, instead leveraging the grounded board representation through language-mediated reasoning to identify the best move.

Finally, we outline a future experimental framework to precisely measure whether the chess puzzle improvement stems from the strategic knowledge encoded in Leela's embeddings, or whether a linguistic encoding of the same information would yield equivalent results.

In summary, we make the following contributions:

[1] LeelaChessZero is an open-source implementation of AlphaZero by Pascutto *et al*. [2019]

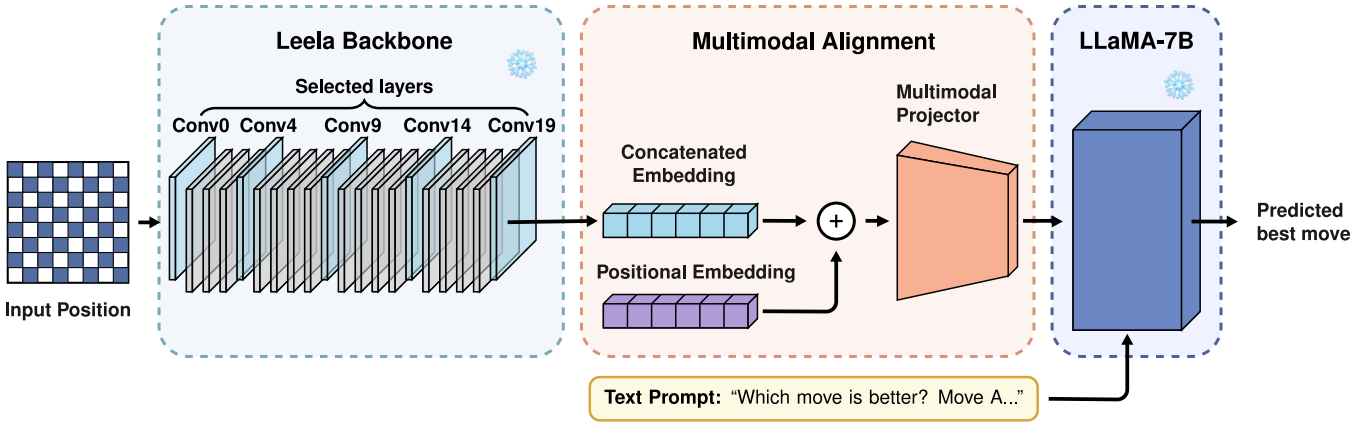

Figure 1: Proposed chess multimodal architecture. We extract and concatenate selected convolutional layers from a frozen LeelaChessZero backbone, bypassing MCTS. This grounded representation is combined with positional embeddings, mapped by a trainable Multimodal Projector and injected into a frozen LLaMA-7B model alongside a text prompt to predict the best MATE move.

- **A Novel Neurosymbolic Architecture** We propose a multimodal integration paradigm that projects the continuous embeddings of an expert chess system (LeelaChessZero [Pascutto, Gian-Carlo and Linscott, Gary, 2019]) into the language space of LLaMA-7B [Touvron *et al.*, 2023]. Our architecture can be naturally extended to other domains that feature expert systems and suffer from high reporting bias.
- **Preliminary Evidence of Grounded Reasoning** We showcase how our architecture yields up to a 34 percentage point increase over the text-only baseline in puzzle accuracy. Crucially, we provide early evidence that the LLM does not merely relay policy data from the embeddings, but rather combines the sub-symbolic board representation with language-mediated reasoning to achieve these gains.
- **A Framework for Isolating Conceptual Grounding** As Work In Progress, we introduce a novel experimental pipeline designed to pinpoint the effect of concept representation: we aim to test whether presenting the *same* concept, either in language or as a sub-symbolic representation, meaningfully changes how well large language models reason.

2 Related work

As we will make extensive use of specific chess terminology throughout this and the next sections, we refer the reader to Appendix A for their definitions.

Improving LLMs' chess performance. Kolasani *et al.* [2025] have shown that Large Language Models significantly struggle in the game of chess, often failing to maintain a coherent board state and hallucinating illegal moves, leading to extremely low win-rates. In an attempt to improve their playing strength, researchers have proposed different fine-tuning approaches, ranging from supervised fine-tuning (SFT) to Reinforcement Learning from Verifiable rewards (RLVR) [Hwang *et al.*, 2025]. For instance, Wang *et al.* [2025] propose an extensive chess puzzle dataset called MATE, which is used to fine-tune LLaMA-3-8B [Grattafiori *et al.*, 2024] in a supervised fashion. In each puzzle, the model is presented with exactly two candidate moves and is asked to choose the correct one. They create four training splits, one containing only the correct move with no explanation (called *no-explain*), while the others address the tactical and positional reasoning behind the best move (called *strategy*, *tactics* and *strategy and tactics*). Compared to the *no-explain* baseline, training on the strategic and tactical splits yielded the highest puzzle solving accuracy.

In general, Hwang *et al.* [2025] show that regardless of the fine-tuning approach adopted, the performance increase in puzzle accuracy plateaus well below human-expert level. Consistent with prior work, they attributed this limitation to the model's limited exposure to chess-related data during pre-training, contending that it cannot be overcome by fine-tuning alone.

Reporting bias and language models. *Reporting bias* has been a long-standing issue in the field of NLP: human conversations systematically omit key details that are taken for granted, *i.e.* "common sense" information that is implied by the context. Language models trained on human data incorporate this bias.

Kamath *et al.* [2026] have recently suggested that Vision Language Models (VLM) perform poorly in tasks that require spatial reasoning [Paiss *et al.*, 2023; Liu *et al.*, 2023; Zhao *et al.*, 2023] due to the *reporting bias* in the training data. They show that even web-scale multimodal datasets contain little to no explicit reasoning traces, arguing that traditional scaling laws cannot bridge this gap if the bias in the data is not removed.

Similarly, driven by pragmatic constraints, chess commentary exhibits a distinct reporting bias by omitting both trivial board transitions and commonplace strategic patterns [Jhamtani *et al.*, 2018].

Multimodal fine-tuning. Another branch of NLP research focuses on multimodal fine-tuning to integrate new modalities into pre-trained LLMs. For instance, LLaMA-AdapterV2 [Gao *et al.*, 2023] enables text-only fine-tuning

by prepending trainable tokens to the model context. They also propose a multimodal variant, where a portion of these tokens is provided by a multimodal encoder. While these multimodal applications have mainly focused on physical signals (e.g. images, video, and audio) [Yin *et al.*, 2024], an emerging body of work has explored integrating structural knowledge representation. For instance, He *et al.* [2024] introduced G-Retriever, proposing to integrate the embeddings of a Graph Neural Network trained on Knowledge Graphs. They reported improved retrieval performance and lower hallucination rates over traditional RAG approaches.

Neural networks trained to play chess. With the introduction of AlphaZero, Silver *et al.* [2017] demonstrated how reinforcement learning-based systems could master board games exclusively through self-play. During gameplay, AlphaZero evaluates positions using Monte Carlo Tree Search (MCTS) guided by a neural network. The network's policy head outputs prior probabilities to narrow the search space, while its value head evaluates leaf nodes, eliminating the need for random rollouts. The MCTS aggregates these neural evaluations to determine the final move and evaluation. Because the network serves to guide this search, its raw outputs from a single forward pass provide approximate heuristics that typically diverge from the full MCTS output.

Moreover, McGrath *et al.* [2022] demonstrated that despite learning exclusively through self-play, AlphaZero autonomously discovers most of the high-level chess concepts developed by human experts. Because these strategic principles are encoded directly within its latent space, networks like Leela provide a natural sub-symbolic substrate to ground the reasoning of language models.

Since AlphaZero implementation is not public, Pascutto *et al.* [2019] created LeelaChessZero, an open-source version based on the architecture described by Silver.

3 Methodology

To integrate LeelaChesZero into LLaMA7B we adopt the following three-stage pipeline:

1. Using LLaMA-AdapterV2[Gao *et al.*, 2023], we integrate LeelaChessZero [Pascutto, Gian-Carlo and Linscott, Gary, 2019] as a new chess modality inside LLaMA7B [Touvron *et al.*, 2023].
2. We construct a custom chess position dataset to align the language space to the embedding space of Leela.
3. We fine-tune our model on the MATE dataset, and measure its puzzle accuracy on different splits. We also monitor whether the model simply relays the best move from the engine. To this end, we train a linear probe on the adaptation prompts to decode the best move.

3.1 Integrating LeelaChessZero into LLaMA

We adopt the multimodal variant of the LLaMA-AdapterV2 [Gao *et al.*, 2023] fine-tuning paradigm, integrating a pre-trained version of LeelaChessZero[2], which is based on a 20 layers Convolutional Neural Network, with a policy and value head. Crucially, the pre-trained weights of both LLaMA and Leela remain frozen during all training stages.

For each pair of chess position - commentary, we apply the following pipeline:

1. We perform a single forward pass through Leela, bypassing MCTS, and extract the activations from the 20 convolutional embeddings: $[Conv_0, ..., Conv_{19}]$. All the activations are taken after the $ReLu$ function.
2. We select the following embeddings: $[Conv_0, Conv_4, Conv_9, Conv_{14}, Conv_{19}]$, where $Conv_i \in \mathbb{R}^{8\times8\times256}$, which we concatenate along the channel dimension such that $Conv_{concat} \in \mathbb{R}^{8\times8\times1280}$. We add a learnable 2-dimensional positional embedding to this tensor, flatten it, and pass it through the multimodal encoder of LLaMA-Adapter V2, an 8-layer transformer. We base our selection strategy on McGrath *et al.* [2022], who suggest that earlier layers primarily encode low-level information such as material count, while deeper layers focus on high-level concepts of strategy and tactics. With this selection strategy we ensure LLaMA receives both kind of information. The full architecture diagram is shown in Figure 1.

The model is then trained with standard self-supervision to predict the next token in the text sequence, utilizing the projected chess embeddings as multimodal context.

3.2 Generating a multimodal alignment dataset

We generate a chess dataset containing pairs of chess positions and a textual description of the location of the pieces and their threats. We start from 1×10^8 chess positions drawn from computer chess games and sample 1×10^6 unique positions from them. Using the Python-Chess library [3], we extract for each position the squares occupied by each piece, their threats, whether the King is in check, and the absolute pins that exist on the board. We then procedurally generate natural language comments for each extracted feature, using different pre-made templates to express the same meaning and injecting variety to avert model collapse (e.g., "{side} {piece} is on {sq}"). We then chain them together using randomly sampled connectors, setting an upper limit to their length to reduce the memory requirements during training. We obtain around 13 chained comments per position, producing a final dataset of 13×10^6 comments relating to the 1×10^6 unique positions.

3.3 MATE dataset adaptation

To evaluate the chess-playing capabilities of the model, we adopt the MATE dataset [Wang *et al.*, 2025], where the model is presented with a board position and two candidate moves and must predict the better one of the two.

Crucially, we adapt the dataset to work with our multimodal approach by removing the standard text-based FEN board representation from the input prompt, replacing it with

[2]We choose the following checkpoint ID: `afc29f06124e0fd5e3d6861cf11bb2c815e998d7a02f38df80c60433efbb9e03`, the weights can be downloaded at https://data.lczero.org/files/networks/

[3]https://github.com/niklasf/python-chess

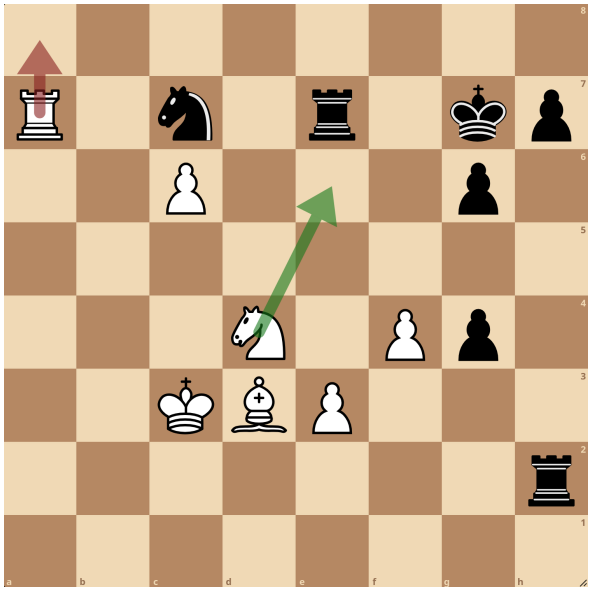

Figure 2: A puzzle from the mate dataset; the corresponding prompt reads: *Which move is better? MoveA:Ra8,tactic Ra8 Nxa8 Trade the lower value piece for a higher value piece. MoveB:Ne6+,tactic Ne6+ Rxe6 Rxc7+ Kh6 f5 Rf6 Correct move: MoveB:Ne6+*

the Leela board encoding. In this way, the model *sees* the board strictly through Leela's representation. Finally, we convert the move notation from the Universal Chess Interface (UCI) notation to the Standard Algebraic Notation (SAN) to preserve consistent encoding across our datasets. In Figure 2 we provide an example of a puzzle in the MATE dataset, together with the textual prompt provided to the model.

3.4 Hyper-parameters selection

We maintain consistent training hyper-parameters across both the multimodal alignment and puzzle fine-tuning phases. Following Gao *et al*. [2023], we optimize the model using Adam [Kingma and Ba, 2015] with a base learning rate of 6.25×10^{-5} and a cosine learning rate scheduler. We train for one full epoch configured to 50 split-epochs, allocating the first 5 split-epochs for learning rate warmup.

4 Experiments

In this section, we outline the experiments we conducted, the empirical results, and our interpretation.

4.1 Modality alignment

First, we align the language space of LLaMA with the embeddings of Leela, adopting the board-description dataset previously introduced: Leela receives the board position with the correct input format, while the language model receives the natural language board description. The model is fine-tuned on these descriptions with self-supervision, training for one full epoch with the hyper-parameters described before. Then, we measure the board decoding accuracy on a held-out test test, iterating through all piece types in each position and prompting the model to generate the square it occupies. The results are reported in Table 1.

Piece	Decoding acc (%)	Evaluated Samples
Pawn	98.97	7093
King	97.67	7240
Queen	96.33	4819
Knight	95.72	4745
Bishop	95.54	5177
Rook	94.64	6423

Table 1: Average decoding accuracy and evaluated samples per piece.

We observe that the model consistently decodes the location of all piece types across the board, showing that the training successfully aligned the chess latent space and the language space.

4.2 Puzzle fine-tuning

We then fine-tune the aligned chess-language model on the four MATE splits, training for one full epoch on each one with the same hyper-parameters as before. We then test the model on a held out test set with 1×10^3 samples: the chess adapter receives the chess position encoded with the Leela input format, while the language model receives the two moves it has to choose between. We generate up to 10 tokens and check if they contain the correct move. As baseline, we also fine-tune a unimodal version without the chess encoder, which receives the board as FEN string, following Wang *et al*. [2025]. Unlike their full-parameter fine-tuning based on LLaMA3-8B, our unimodal baseline uses the original LLaMA-Adapter, a choice that inherently restricts us to LLaMA-7B.

We also investigate whether the adapter simply learns a linear mapping of the target moves, allowing the LLM to lazily relay this information, or if it extracts lower-level features to produce the correct answer autonomously. To this end, we train a linear classifier directly on the tokens produced by the multimodal adapter, which are prepended to LLaMA's context. Our approach is grounded in the Linear Representation Hypothesis [Park *et al.*, 2024], which posits that concepts modeled by a language model can be linearly decoded from its latent space, and the broader NLP probing literature [Belinkov, 2022; Pantazopoulos *et al.*, 2024]. Crucially, because Leela evaluates these positions without MCTS, its embeddings do not linearly encode the final best-move policy (as we show in Appendix B). Consequently, it would be non-trivial for the projector to make this policy information linearly accessible, rendering it an unlikely shortcut that nonetheless should be investigated.

We train a separate probe for each MATE split and evaluate them on a held-out test set that was unseen during both the language model and probe training phases. Complete implementation details are provided in Appendix C.

We present the model puzzle accuracies (unimodal and multimodal) and the relative probe accuracies in Table 2.

First, we observe that our unimodal baseline does not fully replicate the results reported by Wang *et al*. [2025]. Specifically, they noted higher puzzle accuracies in splits containing move explanations compared to the no-explain split. We at-

	Puzzle Acc (%)			Linear Probe (%)	
MATE Split	**Multimodal**	**Unimodal**	**Difference**	**Top-1**	**Top-3**
Mate no explanation	**70.3**	63.5	+6.8	**20.85**	**33.98**
Mate strategy & tactics	**77.0**	60.1	+16.9	14.75	25.52
Mate tactics	**82.3**	48.3	+34.0	16.02	28.04
Mate strategy	**85.8**	64.0	+21.8	18.13	30.58

Table 2: Comparison of LLaMA's end-to-end puzzle accuracy between the unimodal and multimodal versions, and linear probe accuracy on the multimodal adapter embeddings across MATE splits.

tribute this gap to the architectural differences discussed before.

Second, we observe that the multimodal version achieves much higher puzzle accuracies compared to the unimodal one, which is expected given the additional information about the position it extracts from Leela's embeddings. In comparison, the text-only baseline receives no additional context.

Moreover, the multimodal version scores significantly higher on splits that provide a textual explanation of the best move. This suggests that the model leverages these reasoning traces to identify the correct move, rather than merely relying Leela's policy. In fact, the accuracy of the linear probe suggests that the adapter extracts only limited policy data from Leela, as it performs well above random chance (expected accuracy of 5.08×10^{-4}) while remaining well below the overall puzzle accuracy of the model.

Notably, we find that the probe accuracy peaks in the "no explain" split, suggesting that the model relies more heavily on the engine's policy when no linguistic reasoning is provided. Conversely, when natural language explanations are available, the model achieves the highest puzzle accuracy. This suggests a preference for language-mediated strategic reasoning over direct policy extraction.

Nevertheless, it is possible that LLaMA relies on non-linear policy extraction—a limitation we address alongside others in the next section.

5 A Roadmap for Isolating Conceptual Grounding

Our current setup cannot explain *how* the language model integrates linguistic reasoning with Leela's grounded representation, leaving three main questions unaddressed:

1. **Impact of board representation**: is the language model just leveraging the richer geometric representation of the board to identify the best move, largely disregarding the other chess concepts encoded inside Leela?
2. **Impact of chess concept representation**: Is the sub-symbolic representation of chess concepts driving the gains in puzzle accuracy, or would providing the equivalent information in natural language yield comparable results?
3. **Policy leakage**: Is the language model extracting policy data in a non-linear encoding from Leela, explaining the low probe accuracy?

To address these open questions, we propose the following evaluation procedure.

First, to isolate the impact of board representation, we adopt a second chess encoder, ChessLM, developed by Hull [2025]. ChessLM generates rich contextual representations of board states but, crucially, lacks any policy data. We integrate it inside the language model again using LLaMA-Adapter, obtaining two multimodal models and the text only baseline.

Second, we create a dataset pairing board positions with natural language descriptions of their strategic features. These descriptions are procedurally generated based on heuristic evaluations extracted from Stockfish[4] 8 [Stockfish,].

Third, we propose the following fine-tuning pipeline, where at each stage we use the checkpoint produced at the previous stage:

1. **(Multimodal models only)** We warm up the multimodal projectors by training them on the board-description dataset, keeping all other parameters frozen.
2. We fine-tune all three models on the board position dataset; for the multimodal variants, we also train their respective chess projectors.
3. We fine-tune all three models on the chess concept dataset, again training the chess adapters when present. To guarantee that all models have access to equivalent information, we prepend the numerical concept values predicted by Stockfish to the prompts of the text-only and board-only models, without training on them.
4. Finally, we train all three models to play complete chess games, prompting them to reason about the strategic features of the position to produce the best move. Crucially, we freeze the projector weights for the Multimodal Leela model to prevent its policy from leaking into the language model. Following Hwang *et al*. [2025], we suggest training all models using reinforcement learning and dense rewards provided by a teacher model. As in the previous phase, we include the Stockfish concepts in the text-only and board-only prompts to ensure information parity.

We evaluate the resulting models on the Kolasani *et al*.benchmark [2025], where LLMs play against chess en-

[4]Stockfish 8 is a chess engine based on human heuristics that provides numerical evaluations for various strategic chess concepts.

gines of different skill levels and their Elo [Elo, 1978], a measure of their playing strength, is computed (see Appendix A).

In summary, our proposed approach features an additional board encoder devoid of any policy data to isolate the impact of board representation on model performance. It also ensures that all models have access to the same strategic concepts, isolating the impact of their representation on strategic reasoning. Lastly, by freezing the multimodal projector during the game-training phase, we actively prevent the language model from simply probing Leela's embeddings for the best move, ensuring a fair comparison between all models.

6 Conclusion & Future work

Our experiments provide early evidence that a language model can leverage sub-symbolic embeddings to complete symbolic tasks, such as solving chess puzzles, with higher accuracy compared to text-only baselines. In addition, the model combines this sub-symbolic information with language-mediated reasoning to achieve the highest puzzle-solving score.

Building on these findings, we turn to isolating the impact of conceptual representations from other variables, such as board encoding and policy leakage. To measure these dynamics, we are currently deploying the experimental pipeline described in Section 5.

In general, our multimodal architecture can be extended to domains that feature expert networks and suffer from severe forms of reporting bias. Our approach provides a principled mechanism to compensate for these omissions, requiring only a compact alignment dataset to teach the language model how to decode the expert embeddings. This allows the LLM to directly tap into the encoder's domain understanding, bypassing the need for exhaustive linguistic explanations.

Furthermore, future research could combine this multimodal architecture with state-of-the-art training paradigms for reasoning in LLMs —such as Reinforcement Learning from Verifiable Rewards (RLVR) and Chain of Thought [Shao *et al.*, 2024]—to evaluate whether it yields improvements over text-only baselines.

A Chess terminology

Standard Algebraic Notation (SAN): The standard method for recording moves. It identifies the moving piece (K for King, Q for Queen, R for Rook, B for Bishop, N for Knight) and its destination on the 8x8 board (columns 'a'–'h', rows '1'–'8'). For example, "Ke4" indicates the King moves to the e4 square. It also includes symbols for captures (x), checks (+), and checkmate (#).

Universal Chess Interface (UCI): A communication protocol for chess engines. Unlike SAN, it records moves using only the starting and ending square coordinates (e.g., e2e4), omitting piece types.

Chess puzzle: A specific board position where the goal is to find the best sequence of moves to gain an advantage or deliver checkmate.

Absolute Pin: A situation where a piece cannot legally move because doing so would expose its King to a check.

Elo Rating System: A scoring system used to calculate relative skill levels. The expected score for player A with rating R_A against player B with rating R_B is calculated using the logistic curve $E_A = \frac{1}{1+10^{(R_B-R_A)/400}}$. Similarly, the expected score for player B is $E_B = \frac{1}{1+10^{(R_A-R_B)/400}}$ [Elo, 1978].

Forsyth-Edwards Notation (FEN): A single text string that encodes the complete state of a board position, including piece placement, the active player's turn, and castling rights.

Threat: A situation where a piece attacks an opponent's piece, creating the potential for a capture on the subsequent turn if left unaddressed.

B Best move encoding in Leela's embeddings

We also test whether Leela's embeddings linearly encode the correct MATE puzzle solutions. Following the probing approach of Silver *et al.* [2017], we train a sparse linear probe on the *mate strategy and tactics* split, chosen for its diverse set of positions. The complete hyperparameter configurations, dataset splits, and optimization settings are outlined in Table 3. We first perform a grid search over L1 regularization penalties using the validation set to identify the optimal configuration, and subsequently train the final probe on the full training set. For comparison, we also evaluate the same held-out test samples with Leela and MCTS, limiting the search tree expansion to 5×10^3 nodes per position. The results are reported in Table 4.

First, we observe that Leela with full MCTS achieves an accuracy of only 63.06%. Given Leela's superhuman playing strength, this indicates that the target moves in the MATE dataset are not always the optimal choice, but simply the better of the two candidates. We confirm this hypothesis by evaluating 1000 sampled positions, and observe that Leela assigns a higher score to the target move 96.2% of the times. Although the probe is trained on the specific distribution of MATE dataset moves, its accuracy remains well below that of the full MCTS, suggesting that they cannot be linearly decoded from the raw embeddings.

Finally, we observe that earlier layers yield higher decoding accuracy than deeper ones. This likely occurs because early layers preserve raw spatial information tied to specific squares, while deeper layers encode abstract strategic concepts [Silver *et al.*, 2017] that make coordinate recovery difficult.

C LLaMA-Adapter probe

We start by randomly sampling 1×10^5 positions from each split, process them through the multimodal adapter and capture the resulting embeddings, yielding a continuous tensor of shape $\mathbb{R}^{N \times 10 \times 4096}$ (where N is the sample size). We flatten the last two dimensions to obtain a feature matrix of shape $\mathbb{R}^{N \times 40960}$ and apply standard scaling. For the labels, we assign a unique index to all 1968 legal UCI moves and map

Hyperparameter	Leela Embeddings Probe (App. B)	LLaMA Adapter Probe (App. C)
Dataset Split	*strategy and tactics*	Evaluated per MATE split
Grid Train Size	1×10^5	1×10^5
Validation Size	3×10^4	3×10^4
Full Training Size	1.3×10^5	1.3×10^5
Test Size	3×10^4	1×10^4
Regularization	L1: $[1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}, 1 \times 10^{-6}]$	L2: $[1 \times 10^{-1}, 1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 0.0]$
Optimizer	L-BFGS [Liu and Nocedal, 1989]	AdamW [Loshchilov and Hutter, 2018]
Learning Rate	N/A (Line search)	1×10^{-3}
Batch Size	Full batch	2048
Max Iter. / Epochs	100 iterations	50 epochs (w/ early stopping)

Table 3: Hyperparameter configurations and dataset splits for the linear probes.

Layer	Top-1 Acc. (%)	Top-3 Acc. (%)
Layer 0	27.83	45.22
Layer 4	22.22	42.47
Layer 9	21.38	41.06
Layer 14	19.96	39.20
Layer 19	16.72	34.10
Leela + MCTS	**63.06**	**80.58**

Table 4: Top-1 and Top-3 best move decoding accuracy of the linear probe across different layers in LeelaChessZero, compared against the baseline performance of the full Leela + MCTS model.

each puzzle's correct target move accordingly.Given the high dimensionality of the feature space, we utilize a batch approach in PyTorch rather than the L-BFGS optimizer used for Leela's probe. As detailed in Table 3, we optimize the model using AdamW, performing a grid search over L2 regularization penalties on a held-out validation set and adopting early stopping to prevent overfitting. After identifying the optimal weight decay, we retrain the final probe on the combined training and validation set before measuring its Top-1 and Top-3 accuracy on the held-out test set.

References

[Belinkov, 2022] Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, 48(1):207–219, 2022.

[Elo, 1978] Arpad E Elo. The Rating of Chessplayers, Past and Present (Second Edition). *PAST AND PRESENT*, 1978.

[Gao *et al.*, 2023] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. LLaMA-Adapter V2: Parameter-Efficient Visual Instruction Model, 2023.

[Gordon and Van Durme, 2013] Jonathan Gordon and Benjamin Van Durme. Reporting bias and knowledge acquisition. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*, pages 25–30, San Francisco California USA, 2013. ACM.

[Grattafiori *et al.*, 2024] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian. The Llama 3 Herd of Models, 2024.

[He *et al.*, 2024] Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V. Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. *Advances in Neural Information Processing Systems*, 37:132876–132907, 2024.

[Hull, 2025] Ben Hull. Beyond Evaluation: Learning Contextual Chess Position Representations. 2025.

[Hwang *et al.*, 2025] Dongyoon Hwang, Hojoon Lee, Jaegul Choo, Dongmin Park, and Jongho Park. Can Large Language Models Develop Strategic Reasoning? Post-training Insights from Learning Chess. 2025.

[Jhamtani *et al.*, 2018] Harsh Jhamtani, Varun Gangal, Eduard Hovy, Graham Neubig, and Taylor Berg-Kirkpatrick. Learning to Generate Move-by-Move Commentary for Chess Games from Large-Scale Social Forum Data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1661–1671, Melbourne, Australia, 2018. Association for Computational Linguistics.

[Kamath *et al.*, 2026] Amita Kamath, Jack Hessel, Khyathi Chandu, Jena D. Hwang, Kai-Wei Chang, and Ranjay Krishna. Scale Can't Overcome Pragmatics: The Impact of Reporting Bias on Vision-Language Reasoning, 2026.

[Kingma and Ba, 2015] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[Kolasani *et al.*, 2025] Sai Kolasani, Maxim Saplin, Nicholas Crispino, Kyle Montgomery, Jared Quincy Davis, Matei Zaharia, Chi Wang, and Chenguang Wang. LLM CHESS: Benchmarking Reasoning and Instruction-Following in LLMs through Chess, 2025.

[Kwa *et al.*, 2026] Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar,

Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI Ability to Complete Long Software Tasks, 2026.

[Li *et al.*, 2025] Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth Hegde, Kourosh Hakhamaneshi, Shishir G. Patil, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. LLMs Can Easily Learn to Reason from Demonstrations Structure, not content, is what matters!, 2025.

[Liu and Nocedal, 1989] Dong C. Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[Liu *et al.*, 2023] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual Spatial Reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023.

[Loshchilov and Hutter, 2018] Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*, 2018.

[McGrath *et al.*, 2022] Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of Chess Knowledge in AlphaZero. *Proceedings of the National Academy of Sciences of the United States of America*, 119(47), 2022.

[Paiss *et al.*, 2023] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching CLIP to Count to Ten. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3147–3157, Paris, France, 2023. IEEE.

[Pantazopoulos *et al.*, 2024] Georgios Pantazopoulos, Alessandro Suglia, Oliver Lemon, and Arash Eshghi. Lost in Space: Probing Fine-grained Spatial Understanding in Vision and Language Resamplers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 540–549, Mexico City, Mexico, 2024. Association for Computational Linguistics.

[Park *et al.*, 2024] Kiho Park, Yo Joong Choe, and Victor Veitch. The Linear Representation Hypothesis and the Geometry of Large Language Models. In *Forty-First International Conference on Machine Learning*, 2024.

[Pascutto, Gian-Carlo and Linscott, Gary, 2019] Pascutto, Gian-Carlo and Linscott, Gary. Leela chess zero, 2019.

[Shao *et al.*, 2024] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y K Li, Y Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. 2024.

[Silver *et al.*, 2017] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm, 2017.

[Stockfish,] Developers Stockfish. Stockfish. https://github.com/official-stockfish.

[Touvron *et al.*, 2023] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and Efficient Foundation Language Models, 2023.

[Verbeken *et al.*, 2026] Brecht Verbeken, Brando Vagenende, Marie-Anne Guerry, Andres Algaba, and Vincent Ginis. Early Evidence of Vibe-Proving with Consumer LLMs: A Case Study on Spectral Region Characterization with ChatGPT-5.2 (Thinking), 2026.

[Wang *et al.*, 2025] Shu Wang, Lei Ji, Renxi Wang, Wenxiao Zhao, Haokun Liu, Yifan Hou, and Ying Nian Wu. Explore the Reasoning Capability of LLMs in the Chess Testbed. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 611–622, Albuquerque, New Mexico, 2025. Association for Computational Linguistics.

[Yin *et al.*, 2024] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A Survey on Multimodal Large Language Models. *National Science Review*, 11(12):nwae403, 2024.

[Zhao *et al.*, 2023] Tiancheng Zhao, Tianqi Zhang, Mingwei Zhu, Haozhan Shen, Kyusong Lee, Xiaopeng Lu, and Jianwei Yin. VL-CheckList: Evaluating Pre-trained Vision-Language Models with Objects, Attributes and Relations, 2023.

[Zhao *et al.*, 2025] Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. Echo Chamber: RL Post-training Amplifies Behaviors Learned in Pretraining. In *Second Conference on Language Modeling*, 2025.