Enhancing Vision-Language Models for Chess Position Understanding via Symbolic Grounding

Anonymous Authors
Paper under double-blind review

Abstract

Vision-Language Models (VLMs) have shown remarkable capabilities in visual reasoning tasks, yet they struggle with precise spatial understanding required for structured domains like chess. We propose a simple but effective approach that bridges the vision–text modality gap by extracting symbolic representations (FEN notation) from chess board images using a fine-tuned CLIP model. Our method achieves a 75% relative improvement in chess position understanding tasks compared to using vision alone. We demonstrate that providing VLMs with CLIP-extracted FEN strings as textual context significantly improves performance on tasks requiring precise piece localization and board state reasoning. Our findings suggest that hybrid vision–symbolic approaches can enhance VLM performance in structured visual domains.

Keywords: Vision-Language Models, Chess AI, Symbolic Reasoning, CLIP, Multimodal Learning

1 Introduction

Vision-Language Models (VLMs) such as LLaVA and GPT-4V have demonstrated impressive capabilities across diverse visual reasoning tasks. However, they face challenges in domains requiring precise spatial understanding and symbolic reasoning, such as chess position analysis. While humans can easily translate a chess board image into Forsyth-Edwards Notation (FEN) — a standard textual representation — VLMs often struggle with this task due to the fine-grained spatial reasoning required.

Chess has rapidly become a standard testbed for probing the reasoning and planning capabilities of large language models (LLMs) and multimodal systems. Recent work shows that purely text-based LLMs trained on complete game trajectories can reach strong-amateur or even master-level strength when games are represented in FEN or UCI text formats.[1] Similarly, explanation-enriched datasets such as MATE demonstrate that adding strategic and tactical natural-language rationales substantially improves move-selection accuracy, surpassing strong commercial LLMs [3]. At the same time, other work targets human move prediction [4], human-aligned play and skill calibration [9], and purely searchless transformer policies trained on massive supervised datasets [6].

Yet, despite this progress, most chess-LLM work assumes a *textual* representation of the position is already available (PGN, FEN, UCI). In many realistic settings, the input is visual: a rendered board from a GUI, a video frame, or a real-world camera image. In these cases, the bottleneck is not the downstream reasoning but the *vision-to-symbol* mapping that provides a clean, unambiguous board state.

Chess therefore serves as an ideal testbed for studying VLM limitations because:

[1] E.g., ChessLLM reaches ∼1788 Elo against Stockfish when trained on over 20B chess tokens [2].

- **Precise spatial reasoning:** Distinguishing between similar pieces (e.g., white pawn vs. white bishop) and subtle configuration differences requires fine-grained visual understanding.
- **Symbolic grounding:** Chess has a well-defined symbolic language (FEN, algebraic notation) that supports discrete decision-making and formal reasoning.
- **Objective evaluation:** Ground truth board states, legal moves, and engine evaluations can be computed programmatically.

We propose a hybrid vision–symbolic approach where:

- A fine-tuned CLIP-style model [1] is trained to map chess board images into FEN strings in a shared embedding space.
- The extracted FEN is provided as textual context to a strong VLM.
- The VLM performs reasoning tasks (move prediction, position evaluation, state queries) using both visual and symbolic information.

Our approach is instantiated in the open-source *Vichar-CLIP* codebase,[2] which trains a ViT-B/32 CLIP model on chess boards to predict FEN and achieves $> 96\%$ top-1 accuracy on held-out positions.

Contributions. Our contributions are:

- **Novel pipeline:** We introduce, to our knowledge, the first VLM pipeline that explicitly augments vision with CLIP-extracted symbolic representations (FEN) for structured visual reasoning in chess.
- **Empirical validation:** We demonstrate a 75% relative improvement on chess understanding benchmarks compared to a vision-only baseline.
- **Link to chess-LLM literature:** We situate our method within recent work on chess-specific LLMs [2, 12, 5], reasoning with explanations [3], searchless transformers [6], world models [7], and human-aligned engines [9].
- **Open-source:** We release our CLIP training and evaluation pipeline (Vichar-CLIP) and the benchmark scripts necessary to reproduce our results.

2 Related Work

2.1 Vision-Language Models and CLIP

Contrastive vision-language pretraining, as popularized by CLIP [1], learns a joint embedding space for images and text by predicting matching pairs from large-scale web data. This approach has proven remarkably effective for zero-shot classification, retrieval, and transfer across a wide range of visual domains. Subsequent work has focused on making CLIP more efficient (e.g., patch masking and patch selection strategies [14]) and domain-adapting CLIP-like models to specialized settings.

Our method follows the CLIP paradigm but replaces natural-language captions with FEN strings. This mirrors ChessCLIP [12], which jointly embeds game states and language commentary,

[2] `https://github.com/vieveks/vichar_clip_paper_cursor`.

but we focus on high-fidelity *state reconstruction* from images rather than commentary understanding. The Vichar-CLIP repository implements this training loop specifically for chess board images and FEN labels, with variants that also condition on the last move (`FEN+Move`) and achieves $\approx 96.7\%$ top-1 FEN retrieval accuracy on a held-out test set.

2.2 Chess and LLMs

A growing body of work treats chess as a structured testbed for LLM reasoning:

Text-only chess LLMs. ChessLLM [2] fine-tunes an LLM on over 20B tokens of chess games represented in FEN and move text, achieving an Elo of $\sim$1788 against Stockfish and demonstrating that high-quality, full-game supervision significantly improves strength. ChessBench and related work on amortized planning with transformers [6] show that supervised transformers trained on engine-annotated data can reach grandmaster-level performance without explicit search by predicting state- and action-values.

ChessGPT [12] bridges policy learning and language modeling through a large-scale multimodal chess dataset, introducing ChessGPT and ChessCLIP as models that jointly handle game state, policy, and language commentary. The MATE dataset [3] further shows that enriching positions with strategy and tactic explanations yields up to $\sim$95% move-selection accuracy and even boosts the performance of powerful black-box LLMs when explanations are provided as context.

Human-like and human-aligned play. Parallel work focuses on modeling human behavior rather than optimal engine play. ALLIE [9] trains a multi-headed transformer to predict moves, pondering times, and resignation decisions from human game logs, achieving state-of-the-art human move prediction and near-perfect skill calibration (average gap $\sim$49 Elo). Other work explicitly targets move prediction for a single player [4] or subsets of the rating spectrum [15**?**]. These efforts show that LLM-like architectures can capture rich human behavior when provided with accurate textual state representations.

Natural-language chess engines. Smaller models, such as the GPT-2-XL system of Jiang [5], explore training language models to answer natural-language chess queries (e.g., legality, piece identity, evaluation) and to generate moves using instruction fine-tuning and chain-of-thought prompting. While these models exhibit promising qualitative reasoning, their move quality often lags behind specialized engines.

Our work is complementary: we assume a strong, general-purpose VLM or LLM for downstream reasoning but focus on the *visual-to-symbolic* step that almost all prior work sidesteps by assuming FEN or PGN is already given.

2.3 Chess, NLP, and Explanations

Beyond move prediction and engine strength, several works study chess through the lens of NLP:

- **Policy + language integration:** ChessGPT [12] unifies policy learning with language modeling across game records, commentary, and instructions.
- **Sentiment and commentary:** SentiMATE [13] and later ABSA-style models analyze sentiment in chess commentary to evaluate moves. Recent work extends aspect-based sentiment analysis (ABSA) to chess textbooks using player-predicate-move triples and RoBERTa-based classifiers, showing that sentiment over textual descriptions correlates meaningfully with engine evaluations.

- **Human move prediction with LLMs:** Kreiger [4] evaluates GPT-family models and chess-specific engines on predicting human moves, finding that prompt engineering improves LLM performance but chess-specific models such as Maia still dominate.

These works support our claim that *textual* representations of chess positions are powerful interfaces for LLMs. Our goal is to reliably produce such representations from images, so that text-based techniques (sentiment, explanations, policy learning) become available in purely visual settings.

2.4 World Models, Games, and Multi-Agent Reasoning

There is a broader trend of framing interactive environments as sequence modeling problems over text. World-model work in text-based games and synthetic environments demonstrates that transformers can approximate environment dynamics and rewards purely from logged trajectories, then serve as simulators, policies, or planners [8, 7]. WordPlay-style benchmarks and ByteSized corpora treat game state and actions as text, showing strong generalization and planning capabilities in interactive narratives.

In multi-agent settings, Explicit Models of Opponents (EMO) [10] build individual LLM-based opponent models with bi-level feedback, significantly improving win rates and role inference in social deduction games. Meanwhile, narrative-focused frameworks such as PANGeA [11] show how LLMs with memory and validation modules can maintain coherent, rule-abiding narratives in RPGs.

Chess sits at the intersection of these lines of work: it is both a complex planning problem and a social setting where human-aligned reasoning and opponent modeling matter. Our method can be viewed as providing a high-fidelity *symbolic world state* (FEN) from pixels, enabling downstream world-model or multi-agent techniques to operate on structured input.

2.5 Symbolic-Neural Hybrid Systems

Neuro-symbolic AI aims to combine differentiable perception with discrete symbolic reasoning. Prior work in chess includes engines that incorporate sentiment or textual knowledge into evaluation functions [13] and systems that use logic or knowledge graphs to augment neural policies.

Our approach is deliberately simple: we do not perform explicit symbolic search or theorem proving. Instead, we use symbolic representations as an *input modality* to a large VLM. This follows the spirit of recent work that shows explanations and textual structure can scaffold better reasoning in LLMs [3], but applied to a vision-to-symbol pipeline.

3 Method

3.1 CLIP FEN Extraction

We fine-tune a ViT-B/32 CLIP model on a dataset of chess board images paired with their FEN representations. Our implementation is based on the Vichar-CLIP repository, which treats FEN prediction as an image–text retrieval problem: given a board image, the model must select the correct FEN from a candidate set.

Training Objective. We use the standard contrastive loss:

$$\mathcal{L} = -\log \frac{\exp\left(\text{sim}(I_i, T_i)/\tau\right)}{\sum_j \exp\left(\text{sim}(I_i, T_j)/\tau\right)}, \tag{1}$$

where I_i is the image embedding, T_i is the FEN text embedding, $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, and τ is a temperature parameter, following CLIP [1].

We train both a *FEN-only* model and a *FEN+Move* variant in which the text caption concatenates FEN with the last move (e.g., `"<FEN> | e4"`). This mirrors the ChessLLM and ChessGPT practice of including recent move history to improve policy prediction [2, 12].

Inference. Given a test image, we retrieve the most similar FEN from a candidate set using cosine similarity in the joint embedding space. In our experiments we use:

- A *closed set* of candidate FENs drawn from the evaluation benchmark, and
- A *large-pool* setting where the candidate set contains thousands of FEN strings from mixed sources (Lichess puzzles, random game positions).

In both settings, we observe high top-k accuracy (e.g., $> 96\%$ top-1 and 100% top-5 in the closed set configuration).

3.2 VLM Integration

We evaluate two settings:

1. **Visual-only:** The VLM receives only the chess board image.
2. **Visual + FEN:** The VLM receives the image and CLIP-predicted FEN as text context.

Prompt template (with FEN):

> **Question:** [Chess question]
> **FEN representation:** [CLIP-extracted FEN]

In line with explanation-augmented work such as MATE [3], we optionally extend the prompt with additional instructions asking the model to reason about checks, tactics, or strategy, but the core intervention is the inclusion of FEN.

3.3 Evaluation Framework

We design 8 evaluation tasks covering different aspects of chess understanding:

Table 1: Evaluation tasks for chess position understanding.

Task	Type	Example
FEN Extraction	Symbolic	"What is the FEN for this position?"
Piece Count	Counting	"How many pieces does each side have?"
Check Detection	State	"Is either king in check?"
Material Balance	Evaluation	"Who has more material?"
Best Move	Strategic	"What is the best move?"
Tactical Patterns	Reasoning	"Describe any tactical patterns."
Castling Rights	Rules	"Can White castle kingside?"
Piece Localization	Spatial	"What piece is on e4?"

Scoring: We use GPT-4o-mini as an LLM judge to score VLM responses on a 0–1 scale based on correctness compared to ground truth. This follows automated-judge setups used in ChessGPT [12] and other chess-LLM benchmarks.

4 Experimental Setup

4.1 Datasets

Training. For CLIP, we sample ∼100K chess positions from online game databases (e.g., Lichess) and puzzle sets, render them as 2D board images, and pair each with its FEN representation. This setup is analogous to the board-state corpora used in ChessBench [6] and ChessGPT [12], but we only require a single FEN per image, not full move trees or value annotations.

Evaluation. We construct a held-out test set of 12.5K positions stratified across task types (tactics, quiet positions, endgames). For some analyses we subsample task-specific sets (e.g., check detection, best-move puzzles) to mirror the evaluation structure in MATE [3] and natural-language chess engines [5].

4.2 Models

- **CLIP:** ViT-B/32 fine-tuned for 20 epochs using the Vichar-CLIP codebase.
- **VLM:** GPT-4o (OpenAI) used in image+text mode for question answering.
- **Judge:** GPT-4o-mini for automated scoring.

4.3 Baselines

- **Visual-only:** VLM with image input only.
- **Visual + FEN:** VLM with image and CLIP-extracted FEN.

Where appropriate, we compare our symbolic grounding gains qualitatively to those reported in explanation-augmented setups (e.g., the gain from adding strategy+tactic text in MATE [3]) and to purely textual FEN-based reasoning in ChessLLM [2] and ChessGPT [12].

4.4 Implementation Details (Vichar-CLIP)

Training and inference are implemented using the `open_clip` library and PyTorch, following the scripts in `train_clip.py` and `inference_clip.py` from the Vichar-CLIP repository. Training uses:

- Batch size 128, learning rate $1 \cdot 10^{-4}$, AdamW optimizer.
- Mixed-precision (`--fp16`) and gradient accumulation for efficiency.
- Early stopping based on validation retrieval accuracy.

Inference retrieves the top-k FEN candidates and passes the top-1 (or full list) to the VLM as text context.

5 Results

5.1 Main Results

Key Findings.

- FEN context provides substantial improvements across most tasks, similar in spirit to how strategy+tactic explanations improve move selection in MATE [3].

Table 2: Main results on chess understanding tasks.

Metric	Visual-Only	Visual + FEN	Improvement
Average Score	0.250	0.438	+75.0%
Accuracy (≥ 0.9)	12.5%	25.0%	+100%

- The largest gains appear in tasks requiring precise state understanding (check detection, FEN extraction), where a single mislocalized piece can flip correctness.

- Tasks that primarily require coarse material assessment (material balance) see limited gains, consistent with prior observations that LLMs can already count pieces from images reasonably well.

5.2 Per-Task Analysis

Table 3: Per-task performance with and without FEN context.

Task	Visual-Only	+ FEN	Δ
FEN Extraction	0.20	0.50	+150%
Check Detection	0.00	1.00	$+\infty$
Material Balance	1.00	1.00	0%
Piece Count	0.00	0.20	+20%
Castling Rights	0.80	0.80	0%
Best Move	0.00	0.00	0%

Analysis. Check detection shows perfect accuracy with FEN (1.0 vs. 0.0 without), suggesting VLMs struggle with complex piece interactions from vision alone but can exploit symbolic state, much like engine-based or explanation-augmented systems. Material balance performs well in both settings, indicating VLMs can count pieces visually. Best move remains challenging even with FEN, echoing findings from ChessLLM [2], ChessGPT [12], and GPT-2-based natural-language engines [5] that strong policy play requires more than accurate state estimation.

6 Discussion

6.1 Why Does FEN Help?

We hypothesize three mechanisms:

- **Disambiguation:** FEN resolves visual ambiguities (e.g., “White pawn on e4” vs. “White bishop on e4”) and removes reliance on fine-grained pixel-level cues.

- **Grounding:** Textual FEN activates the VLM’s chess knowledge from language pretraining and from chess-specific finetuning in related work [2, 12, 3].

- **Complementarity:** Vision confirms the board layout while FEN provides precise positions. In cases where CLIP mispredicts, the VLM can still cross-check against the image.

This is consistent with evidence that language explanations and structured text scaffolds improve reasoning in chess [3] and beyond [7].

6.2 Relation to Chess-Specific LLMs

Our pipeline can be viewed as a *front-end* for existing chess-LLM systems:

- Text-only engines such as ChessLLM [2], ChessGPT [12], ALLIE [9], and searchless transformers [6] could all operate on CLIP-extracted FEN instead of assuming perfect textual input.
- Human-move prediction models [4] and Maia-style engines [15] could use our visual interface to predict moves from board screenshots or video feeds.
- ABSA-based evaluation methods [13, 16] and sentiment-based engines could be applied to commentary anchored on CLIP-derived FEN in broadcast or educational settings.

In that sense, our work complements policy- and explanation-centric research by solving the "vision bottleneck" that prevents these methods from operating directly on images.

6.3 Limitations

- **CLIP accuracy:** Our CLIP model is not perfect; errors in FEN prediction propagate to the VLM. In critical tactical positions, a single mislocated piece can completely change the evaluation.
- **Task dependency:** Gains are task-specific; strategic reasoning and best-move selection remain challenging even with perfect state information, consistent with prior findings [6, 5].
- **Generality:** We evaluate on standard chess. Extensions to chess variants, 3D boards, or noisy real-world camera feeds would require domain adaptation.
- **Symbolic-only bias:** By privileging FEN, we may underutilize visual cues (e.g., clock, user interface elements) that could be informative in some applications.

6.4 Broader Impact and Connections to World Models and Games

Our results suggest that symbolic intermediate representations can bridge the gap between vision and language in structured domains. This is aligned with broader efforts to build LLM-based world models that learn environment dynamics from logs and use them for planning and simulation [7, 8]. In game and narrative settings, frameworks such as PANGeA [11] and EMO [10] demonstrate that LLMs with memory, validation, and opponent modeling can handle complex multi-agent and narrative dynamics when provided with structured state signals.

Potential applications include:

- **Medical imaging:** Extracting structured patient data (e.g., lesion locations) before diagnostic reasoning.
- **Robotics:** Converting visual scenes to symbolic state descriptions for planning.
- **Education and coaching:** Building explainable tutors that reason over both board images and symbolic annotations.

7 Conclusion

We presented a simple yet effective approach for enhancing VLM performance on chess understanding tasks by incorporating CLIP-extracted FEN representations. Our method achieves a 75% relative improvement over vision-only baselines and plugs directly into the rapidly growing ecosystem of chess-specific LLMs and world-model approaches.

Key Takeaways.

- Symbolic grounding (via FEN) significantly improves VLM spatial reasoning in structured domains like chess.
- Hybrid vision–symbolic systems outperform pure vision or pure text in tasks that require precise state understanding.
- Fine-tuned CLIP models are effective and practical tools for extracting structured representations from images, as demonstrated by Vichar-CLIP.

Future Work.

- Integrate our visual front-end with strong text-only chess engines (ChessLLM, ChessGPT, ALLIE) to evaluate full-game strength from images.
- Extend to other structured domains (Go, molecular structures, circuit diagrams) where a concise symbolic representation exists.
- Combine visual symbolic grounding with explanation-augmented reasoning [3] and multi-agent opponent modeling [10].

References

[1] A. Radford et al. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*, 2021.

[2] Y. Zhang et al. Complete Chess Games Enable LLM Become A Chess Master. *arXiv:2501.17186*, 2025.

[3] S. Wang et al. Explore the Reasoning Capability of LLMs in the Chess Testbed. *arXiv:2411.06655*, 2024.

[4] B. Kreiger. Predicting Human Chess Moves with Large Language Models. Master's Thesis, University of South Florida, 2024.

[5] B. Jiang. Building a Natural Language Chess Engine with Pretraining and Instruction Fine-Tuning. Stanford CS224N Project Report, 2023.

[6] Google DeepMind. Amortized Planning with Large-Scale Transformers (Searchless Chess). *arXiv:2402.04494*, 2024.

[7] K. Xie et al. Making Large Language Models into World Models with Precondition and Effect Knowledge. In *COLING*, 2025.

[8] R. Wang et al. Can Language Models Serve as Text-Based World Simulators? *arXiv:2406.06485*, 2024.

[9] Y. Zhang et al. Human-aligned Chess with a Bit of Search. *arXiv:2410.03893*, 2024.

[10] X. P. Yu, W. Zhang, and Z. Lu. LLM-Based Explicit Models of Opponents for Multi-Agent Games. In *NAACL*, 2025.

[11] S. Buongiorno et al. PANGeA: Procedural Artificial Narrative using Generative AI for Turn-Based Video Games. *arXiv:2404.19721*, 2024.

[12] X. Feng et al. ChessGPT: Bridging Policy Learning and Language Modeling. *arXiv:2306.09200*, 2023.

[13] I. Kamlish, I. B. Chocron, and N. McCarthy. SentiMATE: Learning to play Chess through Natural Language Processing. *arXiv:1907.08321*, 2019.

[14] G. Pei et al. Seeing What Matters: Empowering CLIP with Patch Generation-to-Selection. In *CVPR*, 2025.

[15] J. McIlroy-Young et al. Aligning Superhuman AI with Human Behavior: Chess as a Model System. In *KDD*, 2020.

[16] A. Author et al. Aspect-Based Sentiment Analysis for Chess Move Evaluation from Textbooks. Technical report, 2024.

[17] V. Padman. Vichar-CLIP: Chess Position Identification. GitHub Repository, `https://github.com/vieveks/vichar_clip_paper_cursor`, 2025.

A Benchmark Details

Question Examples

- Q1: What is the FEN (Forsyth-Edwards Notation) for this chess position?
  Ground Truth: `r3k2r/ppb2p1p/2nqpp2/1B1p3b/Q2N4/7P/PP1N1PP1/R1B2RK1`
- Q2: Is either king in check?
  Ground Truth: No
- Q3: What is the best move in this position?
  Ground Truth: Qh2+ (from engine analysis)

B Implementation

Code and benchmarks will be made available at an anonymous repository after the review period. The current implementation is based on Vichar-CLIP for CLIP training and a simple Python harness for VLM evaluation.

C Qualitative Examples

See supplementary materials for detailed VLM responses with and without FEN context, including examples of success and failure cases.