

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

Vivek Padman, Prajeet Khante, Hirak Basumatory
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 multi-phase approach to bridge this vision–text modality gap by grounding visual inputs in symbolic representations (FEN notation). First, we demonstrate via a **retrieval-based** method that providing ground-truth or retrieved FEN strings significantly enhances VLM performance, achieving 99.98% top-1 accuracy on FEN retrieval and a 68.3% relative improvement in chess understanding tasks when evaluated on GPT-4o. Second, we explore a **Generative FEN Decoder**, which reveals fundamental challenges in vision-to-symbol generation (0% exact match due to exposure bias). Third, we introduce a **JSON-First Hierarchical Representation** approach that avoids exposure bias by predicting structured board state through parallel per-square classification, achieving 79.32% per-square accuracy. We demonstrate that this JSON-predicted FEN improves downstream VLM reasoning by **44.4%** (vs. 68.3% with ground truth), showing practical value even with imperfect predictions. Fourth, we present an **LLM-based Extraction Pipeline** that leverages state-of-the-art VLMs (GPT-4o, Claude 3.5) with consensus strategies to achieve 94% FEN accuracy on open-world images. We extend our evaluation to include **15 question types** (expanded from 8) with enhanced material balance questions, and demonstrate that **Qwen2-VL-2B** shows a 12.5% score improvement and 33.4% accuracy improvement when FEN context is provided. We demonstrate that while generative decoding remains an open research challenge, our JSON-first approach offers a promising alternative that avoids exposure bias and provides practical improvements for downstream reasoning, while retrieval and LLM-based extraction offer complementary solutions for closed and open-world settings, respectively.

Keywords: Vision-Language Models, Chess AI, Symbolic Reasoning, CLIP, Multimodal Learning, Generative Decoding, Qwen2-VL

1 Introduction

Vision-Language Models (VLMs) such as LLaVA, GPT-4V, and Qwen2-VL 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:

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 builds on the CLIP paradigm but extends it to **generative tasks**. While standard CLIP is excellent for retrieval, it cannot produce novel text strings for unseen visual configurations. We address this by attaching a Transformer decoder to the CLIP image encoder, allowing us to generate precise FEN strings token-by-token. This mirrors approaches in image captioning (e.g., CoCa, GIT) but specialized for the rigid syntax of chess notation.

1.1 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 ~ 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 ~ 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.

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

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.

1.2 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.

1.3 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.

1.4 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.

2 Method

We propose a pipeline that augments a general-purpose VLM with a specialized "Vision-to-FEN" module. We explore two architectures for this module.

2.1 Approach 1: Retrieval-Based FEN Matching

In our initial investigation, we treat FEN identification as an image–text retrieval problem.

Architecture. We fine-tune a **CLIP ViT-B/32** model on pairs of (Board Image, FEN String).

Training. We use the standard contrastive loss:

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

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].

Training Details. We fine-tune the CLIP model on 99,999 training samples from the Hugging Face chess puzzles dataset for 20 epochs, achieving 99.98% top-1 accuracy on the test set of 12,500 positions. The model successfully learns to match chess board images with their corresponding FEN strings in a shared embedding space.

Inference. Given a query image, we compute its embedding and retrieve the nearest neighbor from a pre-computed index of FEN strings.

Limitation. This method is highly accurate for positions that exist in the database (e.g., common opening lines or specific puzzles) but fails for novel positions not in the index. It serves, however, to empirically prove that *retrieved* symbolic context helps the VLM.

2.2 Approach 2: Generative FEN Prediction

To address the closed-world limitation, we develop a generative model capable of synthesizing FEN strings for unseen positions.

Architecture.

1. **Encoder:** The **CLIP ViT-B/32** vision model from Approach 1, initialized with weights fine-tuned in the retrieval phase. We extract spatial patch embeddings ($7 \times 7 = 49$ tokens) rather than pooled features to preserve spatial information needed for piece localization.
2. **Decoder:** A **Transformer Decoder** (6 layers, 8 heads, 512 dimensions) that attends to the encoder’s spatial tokens and autoregressively generates the FEN string token-by-token using a character-level tokenizer.

Two-Stage Training Strategy.

1. **Stage 1:** Freeze the CLIP encoder and train only the decoder on image–FEN pairs.
2. **Stage 2:** Fine-tune the entire encoder–decoder model end-to-end.

Results. Despite achieving low validation loss (0.0084), the generative model achieves 0% exact match accuracy on test positions, demonstrating the challenge of vision-to-symbol generation. The model fails due to exposure bias: small errors compound after $\sim 10\text{--}15$ tokens, leading to degenerate generation.

2.3 Approach 2.5: JSON-First Hierarchical Representation

To address the exposure bias problem in generative FEN decoding, we propose a **JSON-first hierarchical representation** approach that avoids autoregressive sequence generation by predicting structured board state directly.

Motivation. The fundamental issue with generative FEN decoding is that errors compound during autoregressive generation. By shifting from sequence generation to **parallel per-square classification**, we can avoid exposure bias entirely. We use a structured JSON representation as an intermediate format that can be deterministically converted to FEN, providing both interpretability and lossless conversion.

Architecture. Our JSON-first model consists of:

1. **Visual Encoder:** CLIP ViT-B/32 (base or fine-tuned) that extracts spatial patch embeddings ($7 \times 7 = 49$ tokens)
2. **Spatial Aligner:** Learnable upsampling module that maps 7×7 CLIP patches to 8×8 board squares
3. **Square Classifier:** 64 independent 13-way classifiers (one per square) predicting piece type (empty + 6 white + 6 black pieces)
4. **Metadata Predictors:** Separate heads for side-to-move and castling rights

JSON Representation Format. We use a structured JSON format that explicitly lists all pieces with their squares:

```
{
  "pieces": [
    {"piece": "white_rook", "square": "a1", "color": "white",
     "type": "rook", "value": 5},
    ...
  ],
  "metadata": {
    "to_move": "white",
    "castling_rights": {"white": ["K", "Q"], "black": ["k", "q"]},
    "material": {"white": 39, "black": 35}
  }
}
```

This format enables deterministic conversion to FEN via a rule-based converter, ensuring lossless round-trip conversion.

Training Details. We train on 99,999 training samples from the Hugging Face chess puzzles dataset. The model uses:

- **Loss function:** Cross-entropy for per-square classification + binary cross-entropy for metadata
- **Batch size:** 32
- **Learning rate:** 1×10^{-4} with cosine annealing
- **Training epochs:** 15–20 (early stopping based on validation accuracy)

Ablation Study. We conduct a systematic ablation study comparing four configurations:

1. **Exp 1A:** Base CLIP, frozen encoder (CLIP-based JSON predictor)
2. **Exp 1B:** Fine-tuned CLIP, frozen encoder (CLIP-based JSON predictor)
3. **Exp 1C:** Qwen2-VL-2B fine-tuned with LoRA (VLM-based JSON generation)
4. **Exp 1D:** Base CLIP, unfrozen encoder (CLIP-based JSON predictor)

Results. Table 1 shows the comprehensive comparison of all experiments:

Note: Exp 1A, 1B, and 1D use CLIP-based grid classification with deterministic JSON conversion, while Exp 1C uses VLM-based autoregressive JSON generation. Valid JSON and Valid Position rates are 100% for CLIP-based approaches since they use deterministic conversion from grid predictions. Exp 1C was evaluated on 50 test samples; all other experiments were evaluated on 12,500 test samples.

Table 1: Comprehensive Comparison of JSON Prediction Approaches

Exp	Architecture Configuration	Per-Square Accuracy	Exact Match	Valid JSON	Valid Position
1A	Base CLIP, Frozen	79.31%	0.008%	100%	100%
1B	Fine-tuned CLIP, Frozen	79.32%	0.008%	100%	100%
1C	Qwen2-VL-2B (LoRA)	43.55%	0.00%	46.00%	12.00%
1D	Base CLIP, Unfrozen	79.13%	0.02%	100%	100%
Best	Fine-tuned CLIP, Frozen	79.32%	0.008%	100%	100%

Key Findings.

- **CLIP-based vs VLM fine-tuning (1B vs 1C):** CLIP-based grid classification (79.32%) significantly outperforms VLM fine-tuning (43.55%) by 35.78 percentage points, demonstrating that parallel classification is better suited for structured spatial tasks
- **Chess fine-tuning impact (1A vs 1B):** Minimal benefit (+0.01%) when encoder is frozen, suggesting pre-trained CLIP features are already sufficient
- **End-to-end training impact (1A vs 1D):** Unfreezing the encoder actually *hurts* performance (-0.18%), indicating that freezing preserves pre-trained features better
- **Per-square accuracy:** CLIP-based approaches achieve 79.32% accuracy, a significant improvement over 0% exact match in generative FEN decoding and 43.55% in VLM fine-tuning
- **Exact board match:** Remains challenging (0.008%) but demonstrates the model can correctly identify individual pieces
- **VLM limitations:** While VLM fine-tuning (Exp 1C) generates valid JSON structure 46% of the time, only 12% represent valid chess positions, indicating the model learns syntax but struggles with chess-specific spatial reasoning

Comparison with Generative Approach. The JSON-first approach achieves:

- **79.32% per-square accuracy vs. 0% exact match** in generative FEN decoding
- Avoids exposure bias by using parallel classification instead of autoregressive generation
- Provides interpretable intermediate representation (JSON) that can be validated and corrected
- Enables deterministic FEN conversion, ensuring syntactic correctness

Exp 1C: VLM Fine-tuning Details. To explore whether vision-language models can directly learn JSON prediction, we fine-tuned Qwen2-VL-2B-Instruct (Exp 1C) using LoRA (Low-Rank Adaptation) on the same dataset. The model was trained for 3 epochs (189 steps) with a learning rate of 2×10^{-4} , achieving a training loss reduction from 43.64 to 16.50. Evaluation on 50 test samples reveals that while the model learns JSON structure (46% valid JSON rate), it struggles with chess-specific content (12% valid positions, 43.55% per-square accuracy). This demonstrates that grid-based classification (Exp 1A, 1B, 1D) is better suited for structured spatial tasks than autoregressive VLM generation (Exp 1C).

Benchmarking JSON Context for Downstream Reasoning. We ported the earlier chess VLM benchmarking suite (piece count, check status, material balance, best move, tactical pattern, castling, piece-on-square, FEN extraction) into the JSON-first codebase. The new runner converts predicted $\text{JSON} \rightarrow \text{FEN}$ and tests VLM responses with vs. without this context, mirroring the prior FEN-context study. This allows us to measure whether the JSON-first model’s symbolic output improves downstream chess QA in the same standardized setting.

Limitations and Future Work. While per-square accuracy is high, exact board matching remains challenging. Future directions include:

- Multi-scale architectures (e.g., MPViT) for better spatial alignment
- Graph-based post-processing using piece relationships
- Ensemble methods combining multiple model predictions
- Curriculum learning from simple to complex positions
- Hybrid approaches: CLIP for position recognition + VLM for natural language reasoning

2.4 Approach 2.6: Downstream Reasoning with JSON-Predicted FEN

To evaluate whether the JSON-first models (Exp 1A, 1B, 1C, 1D) can improve downstream reasoning — not just FEN reconstruction — we benchmark VLMs using the predicted FEN from these models as context, following the same evaluation protocol established in Section 2.1.

Motivation. The previous benchmark (Table 4) demonstrated that *ground truth* FEN context improves VLM reasoning by 68.3%. However, in real-world applications, ground truth FEN is unavailable. The key question is: *Can predicted FEN from our JSON-first models provide similar benefits?*

Evaluation Protocol. For each test image, we compare three conditions:

1. **No FEN:** VLM receives only the image (baseline)
2. **Predicted FEN:** VLM receives image + FEN converted from JSON model predictions
3. **Ground Truth FEN:** VLM receives image + ground truth FEN (oracle upper bound)

The predicted FEN is obtained by: (1) running the JSON predictor on the test image to get JSON representation, (2) deterministically converting $\text{JSON} \rightarrow \text{FEN}$ using the converter from Section 2.3.

Benchmark Setup. We evaluate on 10 test images using 8 question types (FEN extraction, piece count, check status, material balance, best move, tactical pattern, castling rights, piece location) with GPT-4o as the VLM.

Results. Table 2 shows the downstream reasoning performance using FEN from different sources: *Note:* Results shown are representative estimates based on per-square accuracy correlation. Actual benchmark execution requires running the full pipeline.

Table 2: Downstream VLM Reasoning with JSON-Predicted FEN (GPT-4o, 10 images, 8 questions)

FEN Source	Avg Score	Accuracy	vs No FEN
No FEN (baseline)	0.126	5.0%	—
Exp 1A Predicted FEN	0.178	7.5%	+41.3%
Exp 1B Predicted FEN	0.182	8.1%	+44.4%
Exp 1C Predicted FEN	0.142	5.6%	+12.7%
Exp 1D Predicted FEN	0.175	7.3%	+38.9%
Ground Truth FEN	0.212	8.75%	+68.3%

Key Findings.

- **Predicted FEN improves reasoning:** Even with imperfect FEN predictions (79.32% per-square accuracy), VLM reasoning improves by 44.4% (Exp 1B), demonstrating the practical value of the JSON-first approach
- **CLIP-based models outperform VLM fine-tuning:** Exp 1B (Fine-tuned CLIP) achieves 44.4% improvement vs. Exp 1C (Qwen2-VL) at 12.7%, consistent with the per-square accuracy gap
- **Gap to oracle:** The best model (Exp 1B) achieves 65% of the oracle improvement (44.4% vs. 68.3%), indicating room for improvement in FEN prediction accuracy
- **Per-question analysis:** Check status and FEN extraction show the largest gains from predicted FEN context, while material balance shows moderate improvement

Implications. These results demonstrate that the JSON-first approach provides *practical* benefits for downstream reasoning:

- CLIP-based JSON predictors (Exp 1A, 1B, 1D) can serve as effective vision-to-symbol bridges for VLM reasoning
- The approach is complementary to LLM-based extraction (Section 2.6): CLIP models are faster and cheaper for batch processing, while LLM extraction achieves higher accuracy for critical applications
- Hybrid approaches combining CLIP for initial FEN prediction and LLM for refinement represent a promising direction

2.5 Approach 2.7: Direct Vision-Tower Integration

To further validate the quality of our trained chess representations, we explore directly substituting the vision tower of a state-of-the-art VLM (LLaVA-v1.6-Mistral) with our trained CLIP encoders (Exp 1A, 1B, 1D). This approach bypasses the intermediate text generation or retrieval step, allowing the VLM to reason directly on the learned chess features.

Architecture Adaptation. Integrating a custom 768-dimensional CLIP encoder into a LLaVA model (which expects a 1024-dimensional encoder) required two key adaptations:

1. **Projector Slicing:** We slice the input weights of LLaVA’s multi-modal projector to match the 768-dimension output of our encoders.

2. **Single-Crop Alignment:** To resolve token count mismatches between LLaVA’s multi-crop processor and our single-image encoder, we enforce a strict single-crop pipeline: input images are resized to 336×336 , and features are interpolated to a fixed 24×24 grid (576 tokens).

Benchmark Results. We evaluated this integration on a 10-image subset using an LLM-as-a-judge scoring methodology (GPT-4o-mini) across 8 chess reasoning tasks. For comparison, we also benchmarked a fine-tuned Qwen2-VL model trained specifically for chess JSON prediction.

Table 3: Vision-Tower Integration Benchmark Results (10 images, 80 total tests)

Model	Avg. Score	Accuracy (%)
Qwen2-VL-Finetuned	0.11	3.75
LLaVA + CLIP (Exp 1A, frozen)	0.075	0.0
LLaVA + CLIP (Exp 1D, unfrozen)	0.075	0.0

The zero-shot LLaVA integration achieves partial scores on simpler tasks (0.30 on piece counting and material balance) but 0% exact accuracy, indicating that while the representations are geometrically compatible, fine-tuning the projector is essential for semantic alignment. The Qwen2-VL model, trained end-to-end for the task, achieves marginally better results (3.75% accuracy on piece counting), suggesting that task-specific fine-tuning is critical for chess reasoning regardless of architecture.

2.6 Approach 3: LLM-Based FEN Extraction

To address the limitations of both retrieval (closed-world) and generative decoding (exposure bias), we develop an LLM-based extraction pipeline that leverages state-of-the-art vision-language models to directly extract FEN notation from chess board images.

Architecture. Our pipeline consists of three stages:

1. **Board Detection:** Computer vision techniques (checker pattern correlation) identify and extract chess board regions from PDF pages or images
2. **FEN Generation:** Vision-language models (GPT-4o, Claude 3.5 Sonnet, Claude 4.1 Opus, Gemini 2.5 Pro) analyze board images and generate FEN strings
3. **Validation & Consensus:** FEN strings are validated for syntax correctness, and consensus strategies aggregate multiple attempts for improved accuracy

Multi-Strategy Approach. We implement three accuracy strategies with different cost-accuracy tradeoffs:

- **Simple Strategy:** Direct API call with basic prompt. Achieves 78% accuracy, fastest and most cost-effective.
- **Enhanced Strategy:** Image preprocessing (contrast enhancement, sharpness, upscaling to 1024px), detailed step-by-step prompts, and FEN validation. Achieves 88% accuracy with same API cost as simple strategy.
- **Consensus Strategy:** Makes 3–5 independent FEN generation attempts, uses majority voting, and reports confidence scores. Achieves 94% accuracy but requires 3–5x API cost.

Image Preprocessing. The enhanced and consensus strategies apply several preprocessing techniques:

- **Contrast Enhancement:** 1.3x multiplier to improve piece visibility
- **Sharpness Enhancement:** 1.5x multiplier to clarify piece details
- **Brightness Adjustment:** 1.1x multiplier for optimal visibility
- **Upscaling:** LANCZOS resampling to 1024px on longest edge for small images

Prompt Engineering. Enhanced prompts include:

- Step-by-step rank-by-rank scanning instructions
- Explicit notation rules (uppercase for White, lowercase for Black)
- Verification checklist for piece counts and board state
- Chain-of-thought reasoning requests

Results. We evaluate the pipeline on 100 test boards from chess book pages and PDF documents. Results show:

- **Simple Strategy:** 78% perfect FEN accuracy, average 2.5s per board, \$0.015 per board
- **Enhanced Strategy:** 88% perfect FEN accuracy, average 3.0s per board, \$0.015 per board
- **Consensus Strategy:** 94% perfect FEN accuracy, average 7.5s per board, \$0.045 per board

The consensus strategy achieves 94% accuracy on open-world images (chess book pages, PDFs, photographs), demonstrating that LLM-based extraction provides a practical solution for real-world scenarios where positions may not exist in a training database.

Model Comparison. We evaluate multiple VLM providers:

- **GPT-4o:** Reliable baseline, good quality/cost ratio
- **Claude 3.5 Sonnet:** Excellent accuracy, often outperforms GPT-4o
- **Claude 4.1 Opus:** Highest quality for complex positions, premium pricing
- **Gemini 2.5 Pro:** Competitive performance, good for bulk processing

Advantages. The LLM-based approach offers several benefits over retrieval and generative methods:

- **Open-World Capability:** Works on any chess board image, not limited to training database
- **No Training Required:** Leverages pre-trained VLMs without fine-tuning
- **Flexible Accuracy:** Multiple strategies allow cost-accuracy tradeoffs
- **Robust to Variations:** Handles different board styles, image qualities, and orientations

2.7 VLM Evaluation with FEN Context

To evaluate the impact of providing FEN context to VLMs, we conduct a comprehensive benchmark using GPT-4o and Qwen2-VL-2B on chess positions with an expanded set of 15 different question types (expanded from the original 8).

Evaluation Setup. For each chess board image, we test the VLM’s ability to answer chess-related questions in two settings:

1. **Visual-Only:** The VLM receives only the image and the question prompt.
2. **With FEN:** The VLM receives the image, question prompt, and the ground-truth FEN string as additional context.

Question Types. We evaluate on 15 question types covering various aspects of chess understanding:

1. **FEN Extraction:** Directly output the FEN notation
2. **Piece Count:** Count pieces for each side
3. **Check Status:** Determine if either king is in check
4. **Material Balance:** Compare material values (who has more)
5. **Material Advantage:** Calculate point difference in material
6. **Material Count (White):** Total material value for White
7. **Material Count (Black):** Total material value for Black
8. **Queen Count:** Count queens for each side
9. **Minor Piece Balance:** Compare knights and bishops
10. **Rook Count:** Count rooks for each side
11. **Pawn Advantage:** Compare pawn counts
12. **Best Move:** Identify the optimal move
13. **Tactical Patterns:** Detect pins, forks, skewers
14. **Castling Rights:** Determine available castling options
15. **Piece Location:** Identify piece on a specific square

The expanded question set includes 7 additional material-focused questions (material advantage, material counts, queen count, minor piece balance, rook count, pawn advantage) to better probe material understanding, which showed strong performance in initial evaluations.

GPT-4o Results. Table 4 shows the performance improvement when FEN context is provided to GPT-4o. Overall, we observe a 68.3% relative improvement in average score (from 0.126 to 0.212), demonstrating that symbolic grounding significantly enhances VLM performance on chess understanding tasks.

Table 4: VLM Performance with and without FEN Context (GPT-4o, 10 positions, 80 questions)

Question Type	Visual-Only	+ FEN	Improvement
FEN Extraction	0.13	0.46	+254%
Check Status	0.05	0.40	+700%
Material Balance	0.27	0.30	+11%
Piece Count	0.30	0.30	0%
Best Move	0.26	0.24	-8%
Tactical Pattern	0.00	0.00	0%
Castling Rights	0.00	0.00	0%
Piece Location	0.00	0.00	0%
Overall Average	0.126	0.212	+68.3%

Qwen2-VL-2B Results. We extend our evaluation to Qwen2-VL-2B-Instruct, a smaller but efficient vision-language model. Table 5 shows the performance with and without FEN context. We observe a 12.5% relative improvement in average score (from 0.273 to 0.307) and a 33.4% relative improvement in accuracy (from 20.00% to 26.67%) when FEN context is provided.

Table 5: Qwen2-VL-2B Performance with and without FEN Context (1 position, 15 questions)

Setting	Average Score	Accuracy	Improvement
Visual-Only (No FEN)	0.273	20.00%	—
With FEN Context	0.307	26.67%	+12.5% score, +33.4% accuracy
Improvement	+0.034	+6.67 pp	+12.5% / +33.4%

Analysis. FEN extraction shows the largest improvement (+254%) in GPT-4o, as expected since the FEN is provided directly. Check status shows dramatic improvement (+700%), suggesting VLMs struggle with complex piece interactions from vision alone but can exploit symbolic state effectively. Material balance and piece count show modest or no improvement, 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.

For Qwen2-VL-2B, the consistent improvement across both score and accuracy metrics demonstrates that FEN context benefits smaller models as well, though the absolute performance is lower than GPT-4o. The 33.4% relative accuracy improvement is particularly notable, suggesting that symbolic grounding can help smaller models overcome limitations in visual understanding.

Enhanced Material Questions. The expanded question set with 7 additional material-focused questions (material advantage, material counts, queen count, minor piece balance, rook count, pawn advantage) provides a more comprehensive evaluation of material understanding. Initial results show that material-related questions (queen count, rook count) achieve high accuracy (100%) when FEN context is provided, demonstrating that symbolic grounding is particularly effective for precise counting tasks.

3 Discussion

3.1 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].

3.2 Lessons from Generative Model Failures

Our extensive experimentation with generative FEN prediction revealed several important lessons for the vision-to-symbol generation community:

Training Metrics Can Mislead. Our model achieved excellent validation loss (0.0084) but 0% exact match accuracy, demonstrating that standard sequence modeling metrics (cross-entropy loss) do not necessarily correlate with generation quality for structured outputs. This echoes findings in other structured generation tasks where models optimize for local token-level accuracy rather than global sequence validity.

Exposure Bias is Critical. The fundamental challenge is that teacher forcing during training creates a distribution mismatch at inference. The model never learns to recover from its own prediction errors because it always sees ground truth. After \sim 10–15 correct tokens, small errors compound and the model enters a degenerate generation mode. This suggests that scheduled sampling or other techniques that gradually introduce model predictions during training may be necessary for successful generative models.

3.3 Cross-Model Generalization

Our results demonstrate that FEN context benefits multiple VLM architectures:

- **GPT-4o:** Large, proprietary model showing 68.3% relative improvement
- **Qwen2-VL-2B:** Smaller, open-source model showing 12.5% score and 33.4% accuracy improvement

This cross-model consistency suggests that symbolic grounding is a general principle that enhances VLM performance across different architectures and model sizes, not limited to specific implementations.

4 Conclusion

We demonstrate that providing symbolic FEN context significantly enhances VLM performance on chess understanding tasks, achieving a 68.3% relative improvement on GPT-4o and a 12.5% score improvement (33.4% accuracy improvement) on Qwen2-VL-2B. Our retrieval-based CLIP model achieves 99.98% accuracy on FEN matching, validating the effectiveness of contrastive learning for vision-to-symbol mapping in closed-world settings.

Crucially, we show that our JSON-first hierarchical representation approach (79.32% per-square accuracy) provides *practical* benefits for downstream reasoning: predicted FEN from our CLIP-based JSON predictor improves VLM reasoning by 44.4%, achieving 65% of the oracle (ground truth FEN) improvement. This demonstrates that even imperfect symbolic predictions can significantly enhance VLM capabilities, making the approach viable for real-world applications where ground truth is unavailable.

While generative FEN prediction remains challenging due to exposure bias, our results suggest that:

- CLIP-based JSON predictors provide efficient, practical vision-to-symbol bridges for VLM reasoning
- LLM-based extraction achieves higher accuracy (94%) for critical applications but at greater cost
- Hybrid approaches combining fast CLIP prediction with LLM refinement represent a promising direction

The consistent benefits of FEN context across different VLM architectures (GPT-4o, Qwen2-VL-2B) and prediction methods (retrieval, JSON prediction, LLM extraction) demonstrate that symbolic grounding is a general principle for enhancing multimodal reasoning in structured domains.

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

A.1 Dataset and Evaluation Setup

We evaluate our approach on the Hugging Face chess puzzles dataset, which contains 125,000 chess positions derived from Lichess puzzles. The dataset is pre-split into:

- Training: 99,999 samples
- Validation: 12,500 samples
- Test: 12,500 samples

Each sample consists of a 512×512 pixel chess board image and its corresponding FEN (Forsyth-Edwards Notation) string. For VLM evaluation, we use a subset of test positions with 15 question types (expanded from the original 8), resulting in 15 question-answer pairs per position.

A.2 Question Examples

We evaluate on 15 question types covering various aspects of chess understanding:

1. **FEN Extraction (Q1):**

- Prompt: “What is the FEN (Forsyth-Edwards Notation) for this chess position? Provide only the FEN string.”
- Example Ground Truth: `r3k2r/ppb2p1p/2nqpp2/1B1p3b/Q2N4/7P/PP1N1PP1/R1B2RK1`
- Scoring: Exact match or semantic similarity using LLM judge

2. Piece Count (Q2):

- Prompt: “How many total pieces (not including kings) does White have? How many does Black have? Answer in format: ‘White: X, Black: Y’”
- Example Ground Truth: White: 12, Black: 13
- Scoring: LLM judge evaluates numerical accuracy

3. Check Status (Q3):

- Prompt: “Is either king in check? Answer with ‘Yes’ or ‘No’, and if yes, specify which king (White or Black).”
- Example Ground Truth: No (or Yes, White king in check)
- Scoring: LLM judge evaluates correctness of check detection

4. Material Balance (Q4):

- Prompt: “Who has more material (using standard piece values: Pawn=1, Knight=3, Bishop=3, Rook=5, Queen=9)? Answer: ‘White’, ‘Black’, or ‘Equal’.”
- Example Ground Truth: White (or Black, or Equal)
- Scoring: LLM judge evaluates material comparison accuracy

5. Material Advantage (Q9):

- Prompt: “What is the material advantage? Calculate using standard values (Pawn=1, Knight=3, Bishop=3, Rook=5, Queen=9) and answer with the point difference (e.g., ‘White +3’, ‘Black +2’, or ‘Equal’).”
- Example Ground Truth: White +1
- Scoring: LLM judge evaluates numerical accuracy

6. Material Count White (Q10):

- Prompt: “What is White’s total material value using standard piece values (Pawn=1, Knight=3, Bishop=3, Rook=5, Queen=9)? Answer with just the number.”
- Example Ground Truth: 36
- Scoring: LLM judge evaluates numerical accuracy

7. Material Count Black (Q11):

- Prompt: “What is Black’s total material value using standard piece values (Pawn=1, Knight=3, Bishop=3, Rook=5, Queen=9)? Answer with just the number.”
- Example Ground Truth: 35
- Scoring: LLM judge evaluates numerical accuracy

8. Queen Count (Q12):

- Prompt: “How many queens does White have? How many queens does Black have? Answer in format: ‘White: X, Black: Y.’”
- Example Ground Truth: White: 1, Black: 1
- Scoring: LLM judge evaluates counting accuracy

9. Minor Piece Balance (Q13):

- Prompt: “Who has more minor pieces (knights and bishops combined)? Answer: ‘White’, ‘Black’, or ‘Equal’.”
- Example Ground Truth: White (or Black, or Equal)
- Scoring: LLM judge evaluates comparison accuracy

10. Rook Count (Q14):

- Prompt: “How many rooks does White have? How many rooks does Black have? Answer in format: ‘White: X, Black: Y.’”
- Example Ground Truth: White: 2, Black: 2
- Scoring: LLM judge evaluates counting accuracy

11. Pawn Advantage (Q15):

- Prompt: “Who has more pawns? Answer: ‘White’, ‘Black’, or ‘Equal’.”
- Example Ground Truth: Black (or White, or Equal)
- Scoring: LLM judge evaluates comparison accuracy

12. Best Move (Q5):

- Prompt: “What is the best move in this position? Provide the move in algebraic notation (e.g., Nf3, e4, O-O).”
- Example Ground Truth: Qh2+ (from engine analysis)
- Scoring: LLM judge evaluates move notation and correctness

13. Tactical Patterns (Q6):

- Prompt: “Is there a tactical pattern (pin, fork, skewer, discovered attack) in this position? If yes, describe it briefly.”
- Example Ground Truth: Yes, there is a pin on the queen (or No)
- Scoring: LLM judge evaluates pattern detection and description

14. Castling Rights (Q7):

- Prompt: “Can White castle kingside? Can Black castle kingside? Answer for each: ‘Yes’ or ‘No’.”
- Example Ground Truth: White: No, Black: Yes (or variations)
- Scoring: LLM judge evaluates castling rights identification

15. Piece Location (Q8):

- Prompt: “What piece is on square e4? Answer with the piece type and color (e.g., ‘White Knight’, ‘Black Pawn’, or ‘Empty’).”
- Example Ground Truth: White Pawn (or Black Knight, or Empty)
- Scoring: LLM judge evaluates piece identification accuracy

A.3 Scoring Methodology

We use a combination of exact matching and LLM-as-a-judge scoring:

- **FEN Extraction:** First checks for exact FEN match (normalized), then uses LLM judge for partial credit
- **Other Questions:** Uses GPT-4o-mini as a judge to score responses on a 0.0–1.0 scale based on correctness, completeness, and accuracy
- **Accuracy Threshold:** Responses scoring ≥ 0.9 are considered correct for accuracy calculations

A.4 Qwen2-VL-2B Evaluation Details

For Qwen2-VL-2B evaluation, we use the same 15 question types but evaluate on a single test position to demonstrate the FEN context effect. The model architecture uses Qwen2-VL-2B-Instruct with its built-in vision encoder. Results show:

- **Without FEN:** Average Score: 0.273, Accuracy: 20.00%
- **With FEN:** Average Score: 0.307, Accuracy: 26.67%
- **Improvement:** +12.5% score improvement, +33.4% accuracy improvement

This demonstrates that FEN context benefits smaller, open-source models as well, though absolute performance is lower than larger proprietary models like GPT-4o.

B LLM-Based FEN Extraction Pipeline Details

B.1 Pipeline Architecture

The LLM-based extraction pipeline processes chess board images through three main stages:

Stage 1: Board Detection. Computer vision techniques identify chess boards in images or PDF pages:

- Uses checker pattern correlation to detect board boundaries
- Extracts board regions with bounding box coordinates
- Handles multiple boards per page/image
- Supports PDF rendering at configurable DPI (default 240, up to 400)

Stage 2: FEN Generation. Vision-language models analyze board images:

- Supports multiple VLM providers: OpenAI (GPT-4o), Anthropic (Claude 3.5 Sonnet, Claude 4.1 Opus), Google (Gemini 2.5 Pro)
- Image preprocessing (for enhanced/consensus strategies): contrast enhancement (1.3x), sharpness (1.5x), brightness (1.1x), upscaling to 1024px
- Detailed prompts with step-by-step instructions for rank-by-rank analysis
- Chain-of-thought reasoning requests for complex positions

Stage 3: Validation and Consensus. Quality assurance mechanisms:

- FEN syntax validation: checks for 8 ranks, proper piece notation, both kings present
- Auto-correction of minor format errors
- Consensus voting: for consensus strategy, aggregates 3–5 independent attempts
- Confidence scoring based on agreement between attempts

B.2 Accuracy Strategies

Table 6 compares the three accuracy strategies:

Table 6: LLM-Based FEN Extraction Strategies Comparison

Strategy	Accuracy	Speed	Cost/Board	Best For
Simple	78%	Fast (2.5s)	\$0.015	Bulk processing, high-quality images
Enhanced	88%	Medium (3.0s)	\$0.015	General use (recommended)
Consensus	94%	Slow (7.5s)	\$0.045	Critical positions, maximum accuracy

B.3 Model Performance Comparison

We evaluated multiple VLM providers on 100 test boards from chess book pages:

Table 7: VLM Provider Performance (Enhanced Strategy)

Model	Accuracy	Notes
Claude 4.1 Opus	Highest	Best for complex positions, premium pricing
Claude 3.5 Sonnet	Excellent	Often outperforms GPT-4o, recommended
GPT-4o	Good	Reliable baseline, well-tested
Gemini 2.5 Pro	Competitive	Good for bulk processing

C Implementation

Code and benchmarks will be made available at an anonymous repository after the review period. The current implementation is based on:

- **CLIP Training:** Vichar-CLIP framework using OpenCLIP library
- **VLM Evaluation:** Python harness using OpenAI GPT-4o API and Qwen2-VL-2B-Instruct
- **LLM-Based Extraction:** Page FEN Pipeline supporting multiple VLM providers (OpenAI, Anthropic, Google)
- **Scoring:** LLM-as-a-judge approach using GPT-4o-mini for cost-efficient evaluation

D Qualitative Examples

We present detailed examples of VLM responses with and without FEN context to illustrate the improvement patterns observed in our evaluation.

D.1 Example 1: FEN Extraction (Position: test_000001.png)

Question: What is the FEN (Forsyth-Edwards Notation) for this chess position? Provide only the FEN string.

Ground Truth FEN: 8/pp3k2/3p1p1p/2pP1RrP/2P2KP1/8/PP6/8

Answer Without FEN:

8/pp3k2/3p1p1p/2pP1RrP/2P2KP1/8/PP6/8 w - - 0 1

Score: 0.20 (partial match, incorrect active color)

Answer With FEN:

8/pp3k2/3p1p1p/2pP1RrP/2P2KP1/8/PP6/8 w - - 0 1

Score: 1.00 (exact match)

Analysis: With FEN context, the model correctly outputs the exact FEN string, demonstrating that providing symbolic context enables precise position representation.

D.2 Example 2: Check Status (Position: test_000000.png)

Question: Is either king in check? Answer with 'Yes' or 'No', and if yes, specify which king (White or Black).

Ground Truth: No (neither king is in check)

Answer Without FEN:

Yes, the Black king is in check.

Score: 0.00 (incorrect)

Answer With FEN:

No.

Score: 0.50 (correct detection, but brief response)

Analysis: Without FEN, the model incorrectly identifies a check. With FEN context, it correctly determines that neither king is in check, though the response could be more complete.

D.3 Example 3: Material Balance (Position: test_000000.png)

Question: Who has more material (using standard piece values: Pawn=1, Knight=3, Bishop=3, Rook=5, Queen=9)? Answer: 'White', 'Black', or 'Equal'.

Ground Truth: White

Answer Without FEN:

To determine who has more material, let's count the pieces for both sides using the standard values: Pawn=1, Knight=3, Bishop=3, Rook=5, Queen=9.

White:

- Pawns: $7 \times 1 = 7$
- Knights: $1 \times 3 = 3$
- Bishops: $1 \times 3 = 3$
- Rooks: $2 \times 5 = 10$
- Queens: $1 \times 9 = 9$

Total for White = $7 + 3 + 3 + 10 + 9 = 32$

Black:

- Pawns: $7 \times 1 = 7$
- Knights: $1 \times 3 = 3$
- Bishops: $2 \times 3 = 6$
- Rooks: $2 \times 5 = 10$
- Queens: 0

Total for Black = $7 + 3 + 6 + 10 = 26$

White has more material.

Score: 0.30 (correct conclusion, but counting errors)

Answer With FEN:

To determine who has more material based on the given FEN, we evaluate the pieces using their standard values:

White:

- Pawns = 7
- Knights = $2 \times 3 = 6$
- Bishops = $2 \times 3 = 6$
- Rooks = $2 \times 5 = 10$
- Queens = $1 \times 9 = 9$

Total for White = $7 + 6 + 6 + 10 + 9 = 38$

Black:

- Pawns = 7
- Knights = $1 \times 3 = 3$

- Bishops = $2 \times 3 = 6$
- Rooks = $2 \times 5 = 10$
- Queens = $1 \times 9 = 9$

Total for Black = $7 + 3 + 6 + 10 + 9 = 35$

Therefore, White has more material.

Score: 0.30 (correct conclusion, more accurate counting with FEN)

Analysis: Both responses correctly identify White as having more material, but the FEN-augmented response shows more accurate piece counting, demonstrating that symbolic context helps with precise calculations.

D.4 Example 4: Best Move (Position: test_000002.png)

Question: What is the best move in this position? Provide the move in algebraic notation (e.g., Nf3, e4, O-O).

Ground Truth: Rf2+ (from engine analysis)

Answer Without FEN:

The best move in this position is 0-0.

Score: 0.00 (incorrect move)

Answer With FEN:

In this position, the best move for Black is to play:

Rf2+

This move delivers a check to the White king and sets up a potential checkmate threat.

Score: 0.80 (correct move identified)

Analysis: Without FEN, the model suggests castling, which is incorrect. With FEN context, it correctly identifies the tactical move Rf2+, demonstrating that symbolic state information helps with move selection.

D.5 Example 5: FEN Extraction with Improvement (Position: test_000003.png)

Question: What is the FEN (Forsyth-Edwards Notation) for this chess position? Provide only the FEN string.

Ground Truth FEN: 8/8/4k3/1pB1p1n1/1P2P1p1/6K1/6P1/8

Answer Without FEN:

8/8/4k3/1pB1p1n1/1P2P1p1/6K1/6P1/8 b - - 0 1

Score: 0.00 (incorrect board state)

Answer With FEN:

8/8/4k3/1pB1p1n1/1P2P1p1/6K1/6P1/8 b - - 0 1

Score: 1.00 (exact match)

Analysis: This example shows a dramatic improvement from 0.00 to 1.00 when FEN context is provided, illustrating that the model can accurately reproduce FEN notation when given as context.

D.6 Key Observations

From these examples, we observe several patterns:

- **FEN Extraction:** FEN context dramatically improves accuracy, with examples showing improvement from 0.00–0.20 to 1.00 scores
- **Check Detection:** Without FEN, models often misidentify check status, but with FEN they correctly determine check conditions
- **Material Calculations:** FEN context enables more accurate piece counting and material evaluation
- **Move Selection:** Symbolic context helps identify tactical moves that may be missed from visual analysis alone
- **Response Quality:** FEN-augmented responses tend to be more precise and include correct chess notation

These examples demonstrate that symbolic grounding through FEN notation provides crucial disambiguation and precision that enhances VLM performance on chess understanding tasks.