Action Plan: Critical Neurosymbolic Experiments for Publication

To ensure the paper is accepted in a reputed journal, we need to add the following "Metric Analysis" and "Refinement" experiments. These transform the narrative from "low exact match accuracy" to "high semantic robustness."

1. Experiment A: The Semantic Error Test (Stockfish Analysis)

- **Goal:** Redeem the **0.008% Exact Match** score in Experiment 1B (JSON/Parallel model) by proving the errors are strategically minor.
- **Protocol:**
    - Take the ~12,500 **Predicted FENs** from Experiment 1B.
    - Run a Stockfish Engine evaluation (depth 10-15 or 0.1s limit) on both the **Predicted FEN** and the **Ground Truth FEN**.
    - Calculate the **Centipawn Loss (CP Loss):** $| \text{Eval}_{\text{True}} - \text{Eval}_{\text{Predicted}} |$.
- **Success Metric:**
    - If **Average CP Loss < 150** (approx 1.5 pawns), we can claim the model preserves the strategic essence of the position despite visual imperfections.
    - **Action:** Generate a **Histogram of CP Loss** to show that most errors are clustered near zero.

2. Experiment B: The "Neurosymbolic Refinement" Loop (Logic Repair)

- **Goal:** Increase the **Exact Match** and **Validity** percentages by filtering neural noise with symbolic logic (System 2).
- **Protocol:**
    - Implement a **Symbolic Refinement Stage** (post-processing script) on the raw Exp 1B predictions.
    - **Apply Logic Constraints (simulating PyReason/SHACL rules):**
        - Constraint 1: If piece count > 32, remove pieces with lowest confidence scores.
        - Constraint 2: If >1 King of the same color, keep only the highest probability one.
        - Constraint 3: Remove any Pawns on Rank 1 or Rank 8.
- **Comparison:**
    - Report **Accuracy Before Refinement** vs. **Accuracy After Refinement**.
    - *Hypothesis:* This should boost Exact Match from 0.008% to a higher visible number (e.g., 5-10%), validating the "Hybrid AI" approach.

3. Experiment C: Downstream Logic Validation (Optional but Recommended)

- **Goal:** Prove that Symbolic Grounding enables **perfect** reasoning compared to VLM guessing.
- **Protocol:**
    - Instead of asking GPT-4 "Is the King in check?", feed the **Predicted FEN** (from Exp 1B) into a simple logical checker (Python chess.Board.is_check() or Datalog).

    - **Compare Reasoning Scores:**
        - *Baseline:* Visual-Only VLM accuracy on "Check Status" (e.g., 20%).
        - *Our Method:* Predicted FEN + Logic Checker accuracy.
    - *Hypothesis:* Even with imperfect FENs, the Logic Checker will outperform the Visual-Only VLM significantly, proving the value of the pipeline.

Revised Experiment C: Downstream Logic Validation

Datalog is the **formal implementation** of the "simple logical checker" mentioned in the original plan.

- **Goal:** Prove that using a dedicated logical solver (Datalog/Python Chess Rule Checker) on the output FEN is vastly superior to the VLM guessing the answer from the image.
- **Protocol:**
    - Take the **Predicted FENs** (from Exp 1B, ideally after Symbolic Refinement in Exp B).
    - Use a **Datalog** framework (or the built-in check function in python-chess for simplicity, and *call* it the "Symbolic Logic Checker" in the paper).
    - For every FEN, ask the checker: **Is_In_Check(White)** and **Is_In_Check(Black)**.
    - Compare the checker's accuracy against the Ground Truth.
- **Comparison in the Paper:**
    - **Baseline:** VLM (e.g., GPT-4o) accuracy on answering "Is in Check?" (Likely low, e.g., $\sim 20\%$).
    - **Our Method (Symbolic Grounding):** Accuracy of the Datalog/Checker using the FEN predicted by our system. (This will be much higher, proving the pipeline's value.)

Summary for the Paper:

Completing Experiment A is mandatory to fix the "0% accuracy" red flag. Experiment B strengthens the "Neurosymbolic" claim by showing that logic can fix neural mistakes.