Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 813–818
DOI: 10.15439/2025F3227
ISSN 2300-5963 ACSIS, Vol. 43

Multi-Modal Deep Learning with Residual and Structure-Guided Refinement for Chess Puzzle Difficulty Prediction

Junlin Chen
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
junlinchen@link.cuhk.edu.cn

Cenru Liu
Ngee Ann Poly
Singapore
liucenru@gmail.com

Yujie Gao
Melbourne Business School
Australia
yujgao2@student.unimelb.edu.au

***Abstract*—The goal of FedCSIS 2025 Challenge is to build a model to predict the difficulty (measured as Lichess rating) of given chess puzzles. To address this task, we propose a three-stage joint visual–statistical framework for predicting Glicko-based difficulty ratings. In the first stage, a convolutional model based on *MobileNetV2* integrates FEN-rendered board images with structured features, including engine-predicted success probabilities, move count, and piece counts, to generate baseline predictions. The second stage employs LightGBM to perform residual refinement, explicitly learning the residual errors of the baseline predictions to correct systematic biases, particularly for extreme difficulty levels. Finally, a domain-informed refinement adjusts the outputs toward interpretable difficulty estimates derived from failure probability distributions and rating-bucket inflection points.**

Our model ranked 9th in the challenge. Experimental results show that residual refinement and domain-informed adjustment significantly reduce mean squared error compared to the baseline visual–statistical model.

***Index Terms*—chess puzzle difficulty, visual–statistical learning, residual error correction, structure-guided refinement**

I. Introduction

CHESS puzzles—tactical positions requiring one or more precise moves—are essential for chess training and skill assessment. Online platforms such as Lichess employ Glicko -based difficulty ratings to recommend puzzles appropriate to players' skill levels[1]. However, deriving such ratings directly from player performance data is expensive and prone to biases caused by uneven rating distributions, motivating research into automated, feature-based prediction methods.

As FedCSIS 2025 Challenge [2] is the extension of IEEE Big Data Cup 2024 [3] chess puzzle competition, both are organized by the KnowledgePit platform[1], we will mainly cite those applied models with high ranking scores in the previous challenge from the IEEE Big Data Cup 2024 report.

a) Deep Learning Methods: Representation learning has recently dominated those image related AI tasks. Ruta *et al.* [4] used convolutional neural networks (CNNs) to predict difficulty from move sequences. Miłosz and Kapusta [5] introduced Transformer-based architectures that treated puzzles as sequential data, while Omori and Tadepalli [6] combined CNNs and LSTMs to jointly encode move sequences and timing information.

b) Hybrid and Cognitive-Inspired Models: Hybrid methods have shown strong performance in recent two competitions. Woodruff *et al.* [7] integrated pretrained Maia/Leela embeddings with tree-based regressors and won the last challenge. Schütt *et al.* [8] proposed cognitive-inspired neural networks mimicking human problem-solving processes.

In this paper, we propose a three-stage framework for predicting Glicko-based puzzle difficulty ratings, developed for the FedCSIS 2025 Challenge on Predicting Chess Puzzle Difficulty:

- A joint visual–statistical model based on *MobileNetV2[9]* that integrates FEN-rendered board images with structured features, including engine-predicted success probabilities, move count, and piece counts, to generate baseline predictions;
- A LightGBM-based[10] prediction refinement stage that adjusts the baseline predictions to mitigate systematic errors, particularly for underrepresented extreme ratings;
- A domain-informed refinement stage that aligns predictions with interpretable chess heuristics by incorporating failure probability distributions and rating-bucket inflection points.

Additionally, we explore uncertainty estimation by introducing a mask prediction strategy that identifies the most unreliable predictions for targeted evaluation.

The remainder of this paper is organized as follows: Section II describes the competition setup, dataset, and evaluation metrics. Section III details the proposed methodology. Section IV presents the mask prediction strategy for uncertainty estimation. Section V reports experimental results and analysis. Finally, Section VI concludes the paper and outlines future work.

II. FedCSIS 2025 Challenge Description

The FedCSIS 2025 Challenge on Predicting Chess Puzzle Difficulty provided a benchmark for assessing machine learning methods in estimating the difficulty of tactical chess puz-

[1]https://knowledgepit.ai/

zles. Participants were required to predict continuous Glicko-based ratings for each puzzle, derived from large-scale player performance statistics on Lichess. Accurate prediction of such ratings is crucial for adaptive training systems and automated puzzle recommendation.

A. Problem Formulation

Formally, each puzzle x_i is associated with a ground-truth difficulty rating $y_i \in \mathbb{R}$. The objective is to learn a predictive function:

$$f^* : x_i \mapsto \hat{y}_i, \quad \text{with} \quad \hat{y}_i \approx y_i \tag{1}$$

where $\hat{y}_i$ denotes the predicted rating.

B. Dataset and Features

The official dataset consists of a large annotated training set and an unlabeled test set prepared specifically for the FedCSIS 2025 Challenge on Predicting Chess Puzzle Difficulty:

- Training Set: 4,557,000 puzzles, each labeled with a human-derived Glicko-2 difficulty rating and accompanied by engine-computed auxiliary statistics.
- Test Set: 2,235 puzzles, sharing an identical feature structure but without ground-truth ratings; predictions on this set determine the final leaderboard ranking.

Each puzzle is represented by structured information that integrates human annotations, contextual metadata, and engine-estimated success probabilities:

1) Core Descriptors: A unique `PuzzleId`, a Forsyth–Edwards Notation (FEN)[**?**] string encoding the complete board configuration (piece placement, side to move, castling rights, and en passant state), and a Portable Game Notation (PGN) sequence listing the puzzle's solution moves.
2) Human-Performance Annotations (training only): Glicko-2 `Rating`, `RatingDeviation`, `Popularity` (measured by user interactions), and `NbPlays` (number of attempts).
3) Contextual Metadata: Descriptive tags such as `Themes`, hyperlinks to the source games (`GameUrl`), and opening-related tags (`OpeningTags`).
4) Engine-Derived Success Probabilities: 20 probability columns estimating the likelihood of solving the puzzle for different skill brackets, including 10 rapid-mode features (`success_prob_rapid_1050–2050`) and 10 blitz-mode features (`success_prob_blitz_1050–2050`).

The feature sets differ slightly between training and test partitions: the training set provides 32 features (including human-performance annotations), whereas the test set contains 25 features, omitting labels and human-derived statistics. The large size and rich heterogeneity of this dataset provide an opportunity to combine statistical priors and visual-spatial representations for accurate difficulty estimation, while also posing significant challenges in balancing feature importance across modalities.

C. Evaluation Protocol

The official evaluation metric was the Mean Squared Error (MSE) between predicted and ground-truth ratings:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2}$$

where N is the number of test puzzles. Lower MSE indicates closer alignment with human-derived difficulty estimates.

D. Specific Challenges of the Task

Despite the availability of rich engine statistics, the task presents several intrinsic challenges:

- **Non-linear dependencies:** Tactical motifs and player success rates interact in complex, non-linear ways; visually similar positions can vary greatly in difficulty due to subtle move-ordering or hidden tactical resources.
- **Imbalanced rating distribution:** The dataset is slightly skewed toward higher-rated puzzles, with puzzles larger than 1700 Elo constituting the largest portion, whereas very easy puzzles are comparatively underrepresented.
- **Multi-source feature fusion:** Effective models must jointly exploit structured numerical data (success probabilities, move counts) and visual-spatial information from FEN-rendered positions.
- **Absence of player-specific metadata:** Unlike human-performance-based systems, the task relies solely on puzzle-intrinsic features, forcing models to infer difficulty without individualized performance history.

III. Proposed Methodology

This section introduces a three-stage hierarchical framework for predicting chess puzzle difficulty ratings. The pipeline integrates a joint visual–statistical model, a LightGBM-based prediction refinement stage, and a domain-informed refinement stage, designed to combine high-capacity representation learning with interpretable domain knowledge.

A. Overall Pipeline

The pipeline consists of three sequential stages, each designed to progressively refine predictions by addressing different aspects of the task.

Step 1: Joint Visual–Statistical Model. The baseline prediction $\hat{y}_{deep}$ is obtained by integrating visual and structured features. FEN-rendered board images are processed through a convolutional backbone (*MobileNetV2*) to extract positional and tactical representations, while structured features—such as engine-derived success probabilities and move counts—provide statistical priors. The two representations are fused to generate the initial difficulty estimate.

Step 2: LightGBM-Based Prediction Refinement. The baseline model exhibits systematic biases, particularly for puzzles with extreme ratings that are underrepresented in the training distribution. To mitigate these biases, a LightGBM regressor is trained to refine the baseline predictions, leveraging its ability to model non-linear relationships in tabular data and to reveal feature importance for interpretability.

Step 3: Domain-Informed Refinement. The final output is adjusted by aligning $\hat{y}_{refined}$ with a domain-inspired difficulty estimate S, derived from engine failure probabilities and rating-bucket inflection points. A confidence-adaptive weighting strategy regulates the strength of this adjustment, applying stronger corrections only when prediction uncertainty is high.

Overall, the framework integrates data-driven representation learning (Step 1), prediction refinement via statistical modeling (Step 2), and consistency enforcement through domain knowledge (Step 3), forming a hybrid approach well-suited for applications that demand both accuracy and interpretability.

B. Data Preprocessing and Feature Engineering

Each puzzle x_i is described by heterogeneous sources of information that combine structural metadata, human performance indicators, and engine-derived statistics:

- **FEN String:** Encodes the full board state, including piece placement, side to move, and castling rights.
- **PGN Sequence:** Lists the solution moves; the total move count is extracted as an indicator of tactical depth.
- **Engine-Derived Success Probabilities:** Estimates for multiple rating buckets across blitz and rapid time controls, serving as statistical priors of player performance at different skill levels.

The structured feature vector $\mathbf{x}_{struct}$ integrates these components:

$$\begin{aligned}\mathbf{x}_{struct} = \{\ & p_r^{\text{blitz}} \mid r \in R, \\ & p_r^{\text{rapid}} \mid r \in R, \\ & \text{move\_count}, \\ & \text{side\_to\_move}, \\ & \text{white\_pieces}, \\ & \text{black\_pieces}\ \}\end{aligned} \tag{3}$$

where R denotes the set of predefined rating buckets. Additional handcrafted features are defined as:

$$\text{move\_count} = |\text{Moves}(x_i)| \tag{4}$$

$$\text{side\_to\_move} = \mathbb{I}(\text{White to move}) \tag{5}$$

$$\text{white\_pieces} = \sum_{c \in \{P,N,B,R,Q,K\}} \text{count}(c) \tag{6}$$

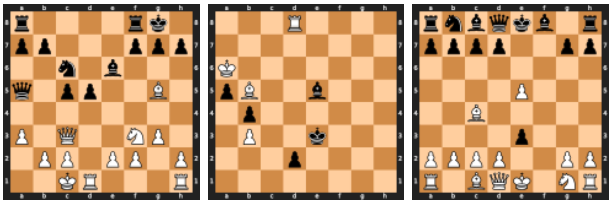
$$\text{black\_pieces} = \sum_{c \in \{p,n,b,r,q,k\}} \text{count}(c) \tag{7}$$

For visual encoding, each FEN string is rendered into a 160×160 RGB image, preserving spatial and tactical motifs not easily captured by numerical features.

Three training samples (puzzles) of PuzzleId,FEN,Moves and Rating fields are illustrated in Table I.

The initial state of the chessboard decoded by FEN of these three training samples (puzzles) with different ratings is illustrated in Figure 1.

Puzzle difficulty ratings range from approximately 400 Elo, corresponding to basic tactical exercises, to over 3000 Elo, representing master-level combinations. The overall distribution is relatively balanced across skill levels, although puzzles rated >=1700 Elo constitute the largest portion (35.3%), followed by intermediate ranges (1000–1399 Elo, 26.0%), as summarized in Figure 2. This mild skew towards higher-rated puzzles may still influence prediction performance for very easy puzzles, which are comparatively less represented.



(a) Rating 1353 (b) Rating 2005 (c) Rating 449

Figure 1. Initial Chess Board State of Different Ratings

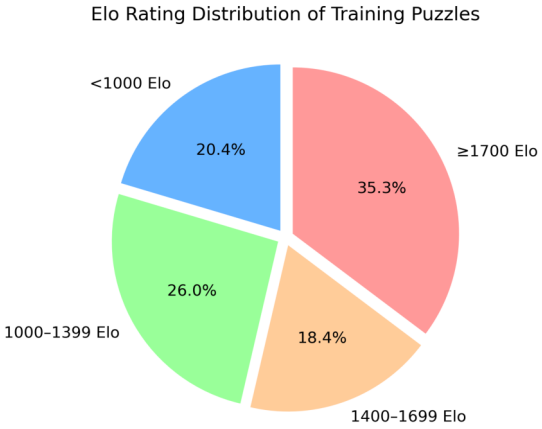


Figure 2. Elo rating distribution of the training set.

C. Stage 1: Joint Visual–Statistical Model

The first stage generates a baseline difficulty estimate $\hat{y}_{deep}$ by jointly encoding visual representations of the chessboard and structured numerical features. This stage serves as the primary representation learning component of the pipeline.

a) Visual Branch: Each puzzle is rendered from its FEN string into an RGB image $x_{img} \in \mathbb{R}^{160\times160}$ MobileNetV2. The image is processed by a convolutional backbone followed by global average pooling (GAP):

$$h_{img} = \phi_{img}(x_{img}) \tag{8}$$

where $\phi_{img}(\cdot)$ maps x_{img} to a compact latent vector $h_{img} \in \mathbb{R}^{d_{img}}$. In our implementation, we adopt *MobileNetV2* pretrained on ImageNet due to its efficiency, achieved through depthwise separable convolutions and inverted residual blocks, which balance accuracy and computational cost.

Table I
THREE SAMPLES OF THE CHESS PUZZLE TRAINING DATA

PUZZLEID	FEN	MOVES	RATING
000Zo	4r3/1k6/pp3r2/1b2P2p/3R1p2/P1R2P2/1P4PP/6K1 w - - 0 35	e5f6 e8e1 g1f2 e1f1	1353
000qP	8/7R/8/5p2/4bk1P/8/2r2K2/6R1 w - - 7 51	f2f1 f4f3 f1e1 c2c1 e1d2 c1g1	2005
0042j	3r2k1/4nppp/pq1p1b2/1p2P3/2r2P2/2P1NR2/PP1Q2BP/3R2K1 b - - 0 24	d6e5 d2d8 b6d8 d1d8	449

b) Feature Branch: Structured statistical features are projected into a dense embedding h_{feat}. Given the feature vector:

$$\mathbf{x}_{struct} = \begin{Bmatrix} p_r^{\text{blitz}},\ p_r^{\text{rapid}} \mid r \in R, \\ \text{move\_count},\ \text{side\_to\_move}, \\ \text{white\_pieces},\ \text{black\_pieces} \end{Bmatrix} \tag{9}$$

$$h_{feat} = \sigma(W_f \mathbf{x}_{struct} + b_f) \tag{10}$$

where $W_f \in \mathbb{R}^{d_f \times |\mathbf{x}_{struct}|}$, $b_f \in \mathbb{R}^{d_f}$, and $\sigma(\cdot)$ denotes the ReLU activation.

c) Fusion and Baseline Prediction: The outputs from the visual and statistical branches, h_{img} and h_{feat}, are integrated to jointly model their complementary information. Rather than treating the two modalities independently, the concatenation of these embeddings allows the network to learn non-linear cross-modal interactions, which are critical for capturing relationships such as how specific positional patterns influence success probabilities.

Formally, the two embeddings are concatenated into a joint representation:

$$h_{joint} = [h_{img}; h_{feat}] \in \mathbb{R}^{d_{img}+d_{feat}} \tag{11}$$

where $[\cdot;\cdot]$ denotes vector concatenation. The joint representation is passed through a fully connected fusion layer to model higher-order dependencies:

$$z = \sigma\left(W_z h_{joint} + b_z\right), \quad z \in \mathbb{R}^{d_z} \tag{12}$$

where $W_z \in \mathbb{R}^{d_z \times (d_{img}+d_{feat})}$ and $b_z \in \mathbb{R}^{d_z}$ are trainable parameters, and $\sigma(\cdot)$ is the ReLU activation introducing non-linearity.

Finally, the fused representation z is mapped to a scalar baseline difficulty estimate through a regression layer:

$$\hat{y}_{deep} = W_o z + b_o, \quad W_o \in \mathbb{R}^{1 \times d_z},\ b_o \in \mathbb{R} \tag{13}$$

The trainable parameters $\{W_z, b_z, W_o, b_o\}$ are optimized to minimize the mean squared error (MSE) between predictions and true ratings:

$$\mathcal{L}_{\text{Stage1}} = \frac{1}{N} \sum_{i=1}^{N} \left(y_i - \hat{y}_{deep,i}\right)^2 \tag{14}$$

where N is the number of training samples.

This fusion design enables the network to exploit both *local spatial patterns* (e.g., piece coordination or king exposure encoded in h_{img}) and *global statistical priors* (e.g., engine-estimated success probabilities and material balance encoded in h_{feat}). By training the fusion weights W_z, the model can learn modality-specific importance and interaction strength, allowing it to prioritize visual or statistical cues depending on the puzzle context. However, the reliance on dense rating regions during training tends to bias $\hat{y}_{deep}$ toward intermediate ratings, highlighting the need for the refinement stages described later.

d) Analysis: The visual branch h_{img} primarily captures spatial and tactical configurations, such as open king exposure, piece coordination, and typical mating patterns, which are essential for modeling human-perceived difficulty. In contrast, the feature branch h_{feat} encodes statistical priors derived from engine-estimated success probabilities and basic material information, providing a complementary perspective based on aggregated player performance.

By concatenating and jointly processing h_{img} and h_{feat}, the model learns to approximate complex non-linear relationships between board patterns and statistical success indicators. The fusion weights W_z implicitly control the relative importance of visual and statistical cues, enabling the network to emphasize modality-specific features depending on the puzzle's structural complexity.

However, the training set is slightly skewed toward high-difficulty puzzles, with puzzles rated ≥ 1700 Elo constituting the largest portion (35.3%), followed by intermediate ranges (1000–1399 Elo, 26.0%). This mild skew can cause the baseline estimate $\hat{y}_{deep}$ to overfit patterns typical of high-rated puzzles, which in turn may lead to overestimation of moderately difficult puzzles and underestimation of very easy ones. Such systematic biases motivate the introduction of a prediction refinement stage (Stage 2) to explicitly adjust for these distributional effects.

D. Stage 2: Prediction Refinement with LightGBM

Although the baseline predictions $\hat{y}_{deep}$ capture both visual and statistical information, they exhibit systematic biases due to the skewed rating distribution and the limited ability of the deep model to generalize across underrepresented rating ranges. To mitigate these biases, we introduce a refinement stage based on residual learning with LightGBM, which explicitly models the prediction error of Stage 1.

a) Residual Definition: For each puzzle i, the residual r_i is defined as:

$$r_i = y_i - \hat{y}_{deep,i}, \quad i = 1, 2, \ldots, N \tag{15}$$

where $r_i > 0$ indicates that the baseline underestimates the difficulty, and $r_i < 0$ indicates overestimation. The residual learning task is formulated as finding a function $f_{lgb}(\cdot)$ that maps an extended feature set to the residuals:

$$f_{lgb} : \mathbf{x}_{lgb,i} \mapsto r_i \tag{16}$$

where the input explicitly combines the baseline prediction and structured statistical features:

$$\mathbf{x}_{lgb,i} = \begin{bmatrix} \hat{y}_{deep,i}, \\ p_r^{\text{blitz}},\ p_r^{\text{rapid}} \ \mid\ r \in R, \\ \text{move count},\ \text{side to move}, \\ \text{white piece count},\ \text{black piece count},\ \ldots \end{bmatrix} \tag{17}$$

b) Residual Prediction: The residual is predicted as:

$$\hat{r}_i = f_{lgb}(\mathbf{x}_{lgb,i}), \tag{18}$$

$$\hat{y}_{refined,i} = \hat{y}_{deep,i} + \hat{r}_i \tag{19}$$

Thus, Stage 2 does not re-learn the entire rating function but only adjusts the baseline estimates by adding the predicted residual, following the principle of boosting-based refinement.

c) Optimization Objective: LightGBM minimizes a regularized differentiable objective:

$$\mathcal{L}_{\text{Stage2}} = \sum_{i=1}^{N} l(r_i, \hat{r}_i) + \Omega(f_{lgb}), \tag{20}$$

where $l(\cdot)$ is typically the squared loss:

$$l(r_i, \hat{r}_i) = (r_i - \hat{r}_i)^2,$$

and $\Omega(f_{lgb})$ controls the complexity of the ensemble of regression trees. Gradient boosting iteratively fits decision trees to the negative gradients of the loss:

$$g_i^{(t)} = \frac{\partial l(r_i, \hat{r}_i^{(t-1)})}{\partial \hat{r}_i^{(t-1)}}, \quad h_i^{(t)} = \frac{\partial^2 l(r_i, \hat{r}_i^{(t-1)})}{\partial (\hat{r}_i^{(t-1)})^2} \tag{21}$$

where $g_i^{(t)}$ and $h_i^{(t)}$ denote the first- and second-order gradients at boosting iteration t, enabling LightGBM to prioritize samples with larger residual errors.

Consequently, the refined predictions $\hat{y}_{refined}$ exhibit reduced systematic errors compared to $\hat{y}_{deep}$, providing a more reliable basis for the final adjustment in Stage 3.

E. Stage 3: Domain-Informed Refinement

The final stage refines the predictions by enforcing consistency with domain-specific heuristics, which approximate human-perceived puzzle difficulty. Unlike the purely data-driven stages, this step introduces an interpretable adjustment based on engine-derived structural signals.

a) Structural Difficulty Estimate: For each puzzle i, a structure-based difficulty estimate S_i is computed from the engine-predicted success probabilities at different rating buckets $r \in R$. The failure probability for bucket r is defined as:

$$\text{fail\_prob}_{r,i} = 1 - \frac{p_{r,i}^{\text{blitz}} + p_{r,i}^{\text{rapid}}}{2}, \tag{22}$$

which approximates the likelihood that a player of rating r fails to solve the puzzle. The structural difficulty estimate S_i is then formulated as the failure-probability-weighted average of rating buckets:

$$S_i = \frac{\sum_{r \in R} \text{fail\_prob}_{r,i} \cdot r}{\sum_{r \in R} \text{fail\_prob}_{r,i} + \epsilon}, \tag{23}$$

where ϵ is a small constant to avoid division by zero. Intuitively, S_i shifts toward higher ratings when high-rated players are more likely to fail, aligning with human intuition that "harder puzzles are those where stronger players also fail."

b) Confidence-Adaptive Refinement: To combine the refined prediction from Stage 2 with the structural estimate, we apply a confidence-adaptive convex combination:

$$\alpha_i = \begin{cases} \alpha_{\text{low}} = 0.2, & \sigma(p_{r,i}^{\text{blitz}}, p_{r,i}^{\text{rapid}}) < 0.04, \\ \alpha_{\text{high}} = 0.3, & \text{otherwise}, \end{cases} \tag{24}$$

$$\hat{y}_{final,i} = (1 - \alpha_i)\hat{y}_{refined,i} + \alpha_i S_i \tag{25}$$

where $\sigma(\cdot)$ denotes the standard deviation of success probabilities across rating buckets. A lower variance implies higher confidence in the baseline prediction $\hat{y}_{refined,i}$ and hence a smaller correction weight (α_{low}), whereas high variance suggests greater uncertainty, allowing stronger reliance on the structural estimate.

c) Analysis: This domain-informed refinement stage provides two key benefits:

- Improved alignment with human intuition: The estimate S_i serves as an interpretable anchor, as it is derived directly from engine-estimated success rates, which converge to human-like difficulty judgments under large-sample conditions.
- Adaptive correction strength: The confidence-based weighting dynamically balances the refined prediction and the structural prior, preventing overcorrection when the model is already confident.

However, the manual tuning of α_{low} and α_{high} limits flexibility, as the optimal weights may vary across rating ranges or puzzle types. Future work may replace these heuristics with a meta-learned or uncertainty-calibrated weighting function $\alpha_i = g_\theta(\mathbf{x}_{struct,i}, \hat{y}_{refined,i})$, where $g_\theta(\cdot)$ can be learned jointly with Stage 2 to further improve consistency and adaptability.

IV. Mask Scoring for Uncertainty Estimation

In addition to the main prediction task, we extend the pipeline to an uncertainty estimation task. The goal is to identify the 10% of test puzzles most likely to be mispredicted. According to the challenge protocol, uncertainty estimation is evaluated by recomputing the mean squared error (MSE) after replacing the masked predictions with ground-truth values. Better localization of uncertain samples yields a lower adjusted MSE, quantified by the N/P ratio, where N is the adjusted MSE and P is the theoretically optimal MSE.

The uncertainty score for puzzle i is computed as:

$$\text{mask\_score}_i = \alpha\, \sigma_{pred,i} \ + \ \beta\, \Delta_i \ + \ \gamma\, M_{outlier,i} \tag{17}$$

where

$$\sigma_{pred,i} = \sqrt{\frac{1}{M} \sum_{m=1}^{M} \left(\hat{y}_i^{(m)} - \bar{y}_i\right)^2}, \tag{18}$$

$$\Delta_i = \left|\hat{y}_{final,i} - \hat{y}_{deep,i}\right|, \tag{19}$$

and there are

- $\sigma_{pred,i}$ measures inter-model variance across M checkpoints, capturing epistemic uncertainty (higher variance $\Rightarrow$ lower confidence);
- Δ_i quantifies the post-correction adjustment magnitude, reflecting disagreement between Stage 1 and Stage 3 (*aleatoric* + *model* uncertainty);
- $M_{outlier,i} \in \{0, 1\}$ flags structural anomalies (e.g., extreme material imbalance);
- α, β, γ are tunable weights calibrated on validation data (default: $\alpha = 1.0, \beta = 0.8, \gamma = 0.1$).

The rationale for combining these three indicators is that no single term sufficiently explains all failure cases:

- $\sigma_{pred,i}$ alone fails when ensembles are consistently biased;
- Δ_i highlights cases requiring strong heuristic correction, often corresponding to difficult puzzles;
- $M_{outlier,i}$ targets rare but structurally atypical puzzles, typically hard for human players as well.

The mask is constructed by ranking samples by mask_score_i.

Using this strategy, we achieved an uncertainty mask ratio of 1.623, ranking 4th among the 9 teams that took part in this additional challenge. Our submitted mask produced a final score of roughly 59,027, whereas a perfect mask would have resulted in about 36,371. The complete rankings and results for all teams will be presented in the [2] paper.

V. Results

A. Experimental Setup

The official training dataset was split into three mutually exclusive subsets with a 7:1:2 ratio, ensuring that each subset preserved the overall rating distribution:

- **Training Set (70%)**: Used to train the Stage 1 deep learning baseline (`Dataset-1`);
- **Validation Set (10%)**: Used for monitoring early stopping of Stage 1 (`Dataset-2`);
- **Residual Learning Set (20%)**: Reserved for Stage 2 LightGBM training (`Dataset-3`), ensuring that refinement was trained only on data unseen by Stage 1.

For Stage 2, LightGBM was trained to predict residuals:

$$r_i = y_i - \hat{y}_{deep,i} \tag{26}$$

using an extended feature set that explicitly included the baseline prediction itself together with structured statistical features, as defined in Eq. 17.

This design allows the residual model to explicitly leverage both the baseline prediction and structured statistical features to correct systematic errors.

Stage 3 applied the domain-informed refinement described in Section III-E, producing the final predictions.

B. Main Results

Table II reports the incremental improvements on the public (preliminary) leaderboard.

Table II
Incremental improvements across stages.

Method	Public MSE	Improvement
Stage 1: Deep Learning (MobileNetV2)	82,266	–
Stage 2: + LightGBM Residual Refinement	72,819	↓ 11.5%
Stage 3: + Domain-Informed Post-Correction	70,408	↓ 14.4% (total)

The final private leaderboard score reached **67,408**, ranking **9th overall**. Both residual refinement and domain-informed post-correction significantly reduced mean squared error (MSE) relative to the baseline.

VI. Conclusion and Future Work

This paper presented a three-stage hybrid framework for predicting chess puzzle difficulty, integrating a joint visual–statistical deep model, residual refinement via LightGBM, and a domain-informed structural correction. The proposed pipeline achieved a top-10 ranking in the FedCSIS Challenge and demonstrated substantial improvements in mean squared error (MSE) compared to deep learning-only baselines. Furthermore, the introduced mask-based uncertainty estimation strategy showed promising capability in accurately localizing unreliable predictions, which is critical for real-world applications such as adaptive puzzle recommendation.

References

[1] Lichess.org, "Puzzle Rating System," Available: https://lichess.org/training/about, accessed Jan. 2025.
[2] J. Zyśko, M. Ślęzak, D. Ślęzak, and M. Świechowski, "FedCSIS 2025 knowledgepit.ai Competition: Predicting Chess Puzzle Difficulty Part 2 & A Step Toward Uncertainty Contests," in *Proc. 20th Conf. Comput. Sci. Intell. Syst. (FedCSIS)*, vol. 43, Polish Inf. Process. Soc., 2025. doi: http://dx.doi.org/10.15439/2025F5937.
[3] J. Zyśko, M. Świechowski, S. Stawicki, K. Jagieła, A. Janusz and D. Ślęzak, "IEEE Big Data Cup 2024 Report: Predicting Chess Puzzle Difficulty at KnowledgePit.ai," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8423-8429, doi: 10.1109/BigData62323.2024.10825289.
[4] D. Ruta, M. Liu and L. Cen, "Moves Based Prediction of Chess Puzzle Difficulty with Convolutional Neural Networks," 2024 IEEE International Conference on Big Data (BigData), Washington, DC, USA, 2024, pp. 8390-8395, doi: 10.1109/BigData62323.2024.10825595.
[5] S. Miłosz and P. Kapusta, "Transformer-based Modeling of Chess Puzzle Difficulty from Sequential Data," arXiv preprint arXiv:2403.01234, 2024.
[6] M. Omori and P. Tadepalli, "Joint Puzzle and Player Rating Estimation with CNN-LSTM Models," in *Proc. IEEE BigData Cup*, 2024.
[7] T. Woodruff, O. Filatov, and M. Cognetta, "The bread emoji team's submission to the IEEE BigData 2024 Cup: Predicting chess puzzle difficulty challenge," in *2024 IEEE International Conference on Big Data (BigData)*, pp. 8415–8422 2024.
[8] A. Schütt, T. Huber, and E. André, "Human Problem-Solving Inspired Neural Networks for Chess Puzzle Difficulty Prediction," in *Proc. IEEE BigData Cup*, 2024.
[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *arXiv preprint arXiv:1801.04381*, 2019. [Online]. Available: https://arxiv.org/abs/1801.04381
[10] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 3146–3154.