Proceedings of the 20th Conference on Computer Science and Intelligence Systems (FedCSIS) pp. 831–835 ISSN 2300-5963 ACSIS, Vol. 43
DOI: 10.15439/2025F7603

Pretraining Transformers for Chess Puzzle Difficulty Prediction

Szymon Miłosz
0009-0007-7408-7304
Institute of Applied Computer Science
Lodz University of Technology
Lodz, Poland
szymonmilosz99@gmail.com

***Abstract*—This paper presents our third-place solution for the FedCSIS 2025 Challenge: Predicting Chess Puzzle Difficulty — Second Edition. Building on our prior GlickFormer architecture, we develop a transformer-based approach featuring a novel multitask pretraining strategy that combines masked-square reconstruction with solution policy prediction. Our spatial-only architecture directly embeds solution moves, eliminating temporal modules, while integrating human-centric priors through Maia-2 engine solve-rate predictions. Evaluated on the Lichess puzzle corpus, our approach reduces validation MSE by 30.4% compared to from-scratch training and achieves competitive results (test MSE: 55.9k) despite distribution shifts in the competition environment. In an auxiliary uncertainty-masking side competition organized post hoc, our simple calibration-sensitivity mask achieved the best score and won the side competition. Details are provided in the Appendix.**

***Index Terms*—Chess AI, Transformer Pretraining, Puzzle Difficulty, Self-Supervised Learning, Glicko-2**

I. Introduction

CHESS puzzles challenge players to identify tactical sequences that yield decisive gains. On platforms like Lichess, puzzle difficulty is quantified via Glicko-2 ratings derived from human solve rates. The requirement for thousands of human attempts makes rapid rating of new puzzles impractical, motivating automated difficulty prediction methods that model human problem-solving rather than engine-optimal play.

Building on our prior GlickFormer (11th place in 2024) [1], we introduce GlickFormer v2, which placed third in the FedCSIS 2025 Challenge: Predicting Chess Puzzle Difficulty — Second Edition. [2] While retaining a transformer backbone, this new version simplifies computation by eliminating explicit temporal modules in favor of embedding future solution moves directly in the input representation. Additionally, we incorporate human-centric priors through solve-rate predictions from Maia chess engines [3] across various Elo bands.

Our core contribution is a novel multitask pretraining strategy inspired by large language models (LLMs), which simultaneously trains the model to reconstruct randomly masked board squares while predicting the puzzle's correct solution move. When fine-tuned on the 4.7 million-position training set, this pretraining significantly outperforms both from-scratch training and masked-only pretraining baselines. To address the distribution shift between validation and test ratings, we implement post-hoc distribution scaling technique used by Woodruff et al. [4].

The resulting model achieves competitive performance despite its simplified architecture, demonstrating that the combination of efficient spatial processing, human-aligned inputs, and targeted pretraining can effectively capture human puzzle-solving complexity without expensive temporal modeling.

II. Related Work

Early approaches to chess puzzle difficulty estimation relied primarily on human-derived ratings, but recent competitions have established new state-of-the-art benchmarks. The 2024 IEEE BigData Cup [5] saw significant advances: Björkqvist et al. [6] combined handcrafted features with engine-extracted signals from Maia, Leela, and Stockfish using a residual neural network and LightGBM to place third. Schütt et al. [7] developed a human problem-solving inspired CNN with auxiliary theme and move-prediction tasks, securing second place. Rafaralahy [8] applied a pairwise learning-to-rank framework using RankNet to simulate puzzle "matches" for Glicko-2 rating inference, achieving fourth place.

The current state-of-the-art was established by the bread emoji team [4], winners of the 2024 competition. Their solution employed an ensemble of pretrained Maia and Leela embedders with distribution rescaling postprocessing, achieving over 13% reduction in MSE compared to runner-up solutions. Our original GlickFormer [1] leveraged a factorized spatio-temporal transformer to model board states and move sequences, placing 11th in 2024 while demonstrating attention mechanisms' potential for capturing human-like puzzle difficulty, though its computational costs were substantial.

Our multitask pretraining approach draws inspiration from LLM strategies. The masked reconstruction component adapts BERT's masked language modeling objective [9] to chess by predicting masked board squares based on spatial context. This technique shares strong parallels with Vision Transformers (ViT) [10], which successfully adapted BERT-style pretraining to image recognition tasks by treating image patches as tokens. The policy prediction task shares similarities with next-token prediction in autoregressive models like GPT [11], but focuses

specifically on identifying optimal tactical moves. This dual-objective approach mirrors multi-task pretraining in models like T5 [12], creating robust representations through diverse objectives. Recent chess-specific transformers like the Chess Transformer [13] have demonstrated the value of language-modeling-inspired approaches for move generation, though not specifically for difficulty prediction.

Building on these foundations, GlickFormer v2 integrates efficient spatial-only transformers with LLM-inspired multitask pretraining and competition-proven calibration techniques. Our approach adapts established language modeling paradigms to structured prediction tasks in chess while building upon the state-of-the-art results from the 2024 competition.

III. Methodology

Building upon our previous GlickFormer architecture, we maintain the core data pipeline and noise-aware target sampling strategy while introducing three key enhancements: improved input representations, a leaner spatial-only transformer backbone, a novel multitask pretraining schedule, and the integration of post-hoc distribution calibration.

A. Data and Stochastic Target Sampling

The training corpus comprises 4.7 million puzzles with Glicko-2 ratings r_i, rating deviations RD_i, complete solution move sequences, and solve-rate predictions from the Maia-2 chess engine [3]. These predictions are provided for multiple Elo bands (approximately 1000-2100, though exact ranges were unspecified by competition organizers), including both rapid and blitz time control configurations. The solution moves represent the optimal tactical sequence required to solve each puzzle, typically ranging from 1-5 moves in length.

To evaluate model performance during development, we reserve 1% of puzzles (approximately 47,000) for validation, selected via stratified sampling across puzzle themes to ensure representativeness.

For pretraining, we construct a 14-channel input representation where 13 binary planes indicate piece occupancies: each plane corresponds to a specific piece type-color combination or empty square, with values set to 1 where present and 0 otherwise. This is supplemented with a binary mask-indicator plane that marks randomly selected squares for masking (1 = masked, 0 = visible), with all other channels zeroed out at masked positions.

During fine-tuning, we use a richer 55-channel input with three components: the same 13 binary board-state planes; 20 binary planes encoding the complete solution line (truncated or zero-padded to exactly 5 moves, each move represented as side $\times$ {from,to} pairs); and 22 continuous-valued Maia-2 solve-rate priors broadcast as constant spatial planes. All channels are linearly projected to form 64 tokens (one per board square) for transformer processing.

For stochastic target sampling, we normalize ratings using dataset statistics:

$$\mu_i = \frac{r_i - \mu_{\text{dataset}}}{\sigma_{\text{dataset}}} \tag{1}$$

$$\sigma_i = \frac{\mathrm{RD}_i}{\sigma_{\text{dataset}}} \tag{2}$$

where μ_{dataset} and σ_{dataset} represent the mean and standard deviation of all puzzle ratings in the training set. We model each rating as a normal distribution $\mathcal{N}(\mu_i, \sigma_i^2)$ and generate noisy targets during mini-batch sampling:

$$y_i \sim \mathrm{clip}\left(\mathcal{N}(\mu_i, \sigma_i^2), \mu_i - 3\sigma_i, \mu_i + 3\sigma_i\right) \tag{3}$$

This injects label noise that regularizes the regression objective while respecting rating deviation bounds.

B. Model Architecture

GlickFormer v2 employs 12 encoder-only transformer blocks with 768-dimensional embeddings and 24 attention heads with Smolgen positional encoding, using Mish activation functions. [14] The model processes different inputs during pretraining and fine-tuning phases: pretraining uses only the 14-channel tensor, while fine-tuning incorporates the extended 55-channel tensor where both the solution-move projector W_{sol} and Maia-prior projector W_{maia} are randomly initialized and trained alongside the backbone.

Each input stack undergoes projection via linear embedding into 64 tokens (one per chess square), which are then processed by the transformer backbone. During pretraining, two specialized heads operate simultaneously: a masked-square classifier implemented as an MLP applied independently to each token predicts piece types at masked positions, while a policy head employs two parallel MLPs to project each token to "from" (query) and "to" (key) vectors. Move logits are computed as dot products between all query-key pairs ($\mathbf{q}_a^\top \mathbf{k}_b$ for move $a \to b$), implementing the Leela Chess Zero policy formulation [15]. This formulation considers only square-to-square transitions without special move types, inherently omitting promotion moves since puzzles rarely involve pawn underpromotions in their solution lines.

For policy prediction during pretraining, we consider only the first move of each puzzle's solution sequence. This design choice reflects that the initial move typically represents the most significant tactical insight distinguishing puzzle difficulty, while subsequent moves are often forced responses or technical conversions contributing less to human solve-rate variance. This simplification maintains predictive power while reducing computational complexity.

For fine-tuning, both pretraining heads are discarded and replaced with the value head architecture from the original GlickFormer. This consists of a linear layer that projects the entire sequence of 64 token embeddings to a hidden dimension, followed by Mish activation and a final linear layer that outputs a scalar difficulty prediction. The entire model - including the transformer backbone, input projectors, and regression head - is trained end-to-end during this phase,

allowing the pretrained representations to adapt while specializing for rating prediction.

C. Multitask Pretraining

Our pretraining approach masks k random squares per puzzle, where $\mathcal{M}$ denotes the set of masked squares. We define two complementary loss functions:

The masked reconstruction loss predicts missing piece types:

$$\mathcal{L}_{\text{msk}} = -\frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \log p_{\text{msk}}(c_i) \tag{4}$$

where c_i is the true piece class at square i.

The policy loss identifies the correct solution move:

$$\mathcal{L}_{\text{pol}} = -\log p_{\text{pol}}(a^\star) \tag{5}$$

where $a^\star$ is the ground-truth solution move.

The combined pretraining objective balances these losses:

$$\mathcal{L}_{\text{pre}} = \lambda_{\text{msk}} \mathcal{L}_{\text{msk}} + \lambda_{\text{pol}} \mathcal{L}_{\text{pol}} \tag{6}$$

Figure 1 illustrates our dual pretraining objectives. The model receives boards with randomly masked squares (Fig. 1a), then must both reconstruct the original pieces at masked positions (Fig. 1b) and predict the optimal solution move indicated by the arrow (Fig. 1c).

Following the Chinchilla scaling rule [16], we train with $20\times$ more tokens than parameters using AdamW optimization with batch size 256, dropout rate 0.1, weight decay 0.1, initial learning rate 5×10^{-4}, 10% linear warmup, and cosine decay to final learning rate 5×10^{-5}. We empirically set $k = 8$ and weighting coefficients $\lambda_{\text{msk}} = 1$, $\lambda_{\text{pol}} = 0.1$ based on validation performance.

D. Fine-Tuning Procedure

During supervised fine-tuning, we employ the 55-channel input tensor with newly initialized projectors for solution-move and Maia-prior planes. The entire model trains end-to-end without freezing components. We discard pretraining heads and replace them with the regression head described in Model Architecture section, minimizing the mean squared error:

$$\mathcal{L}_{\text{ft}} = \frac{1}{N} \sum_{i=1}^{N} (\hat{r}_i - r_i)^2 \tag{7}$$

using AdamW optimization [17] with dropout rate 0.1, weight decay 0.1, batch size 256, and initial learning rate 5×10^{-5}. We reduce the learning rate by a factor of 0.1 after each epoch without validation improvement, with a maximum of two reductions. This configuration maintains consistency with our original GlickFormer implementation while leveraging the enhanced pretrained representations.

E. Post-Hoc Distribution Calibration

To address distribution shifts between validation and test environments, we implement the distribution rescaling technique introduced by Woodruff et al. This calibration corrects systematic biases in predicted ratings caused by the key difference in data collection: limited solving attempts (25-50 per puzzle) compared to the training data, which shifts the rating distribution toward the mean μ by preventing full convergence.

The calibration function is defined as:

$$R(\hat{r}) = \begin{cases} \max\left(\mu, \hat{r} - \alpha \min\left(1, \left(\frac{\hat{r}-\mu}{\delta}\right)^\gamma\right)\right) & \hat{r} \geq \mu \\ \min\left(\mu, \hat{r} + \beta \min\left(1, \left(\frac{\mu-\hat{r}}{\delta}\right)^\gamma\right)\right) & \hat{r} < \mu \end{cases} \tag{8}$$

with empirically tuned coefficients $\alpha = 200$, $\beta = 400$, $\gamma = 2$, $\delta = 550$. This formulation applies progressively stronger corrections to predictions farther from μ, counteracting the observed shift toward the mean caused by limited solving attempts.

IV. Results and Discussion

Our experimental evaluation addresses the effectiveness of our pretraining strategies, GlickFormer v2's performance relative to state-of-the-art baselines, and the impact of post-hoc calibration. All validation results use our stratified validation set while test results report official competition metrics.

A. Evaluation Protocol

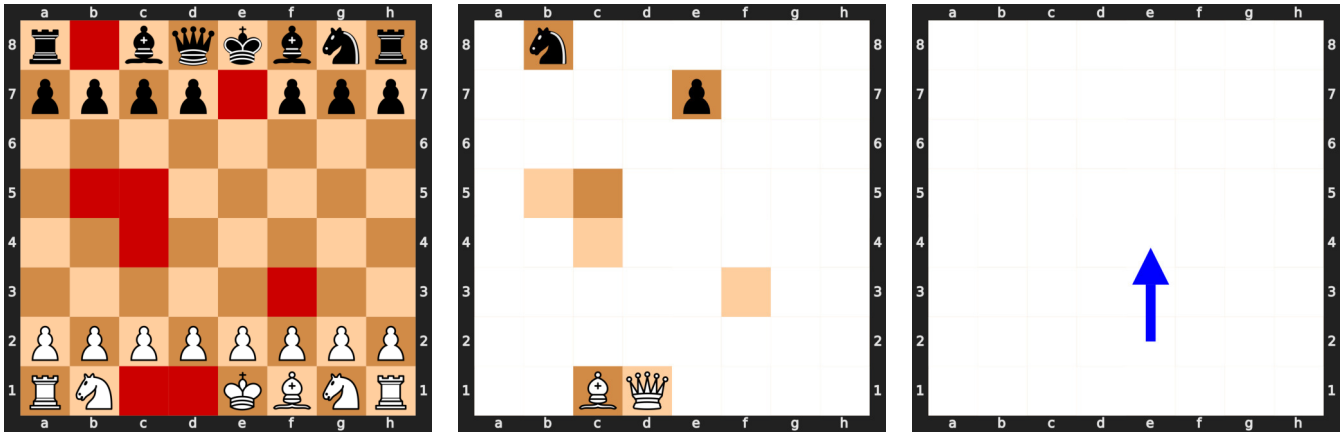
We compare three pretraining strategies: from-scratch training (baseline), masked-only pretraining ($\lambda_{\text{pol}} = 0$), and our multitask approach ($\lambda_{\text{pol}} = 0.1$). Baseline comparisons include GlickFormer v1 [1] and the bread emoji 2024 solution [4]. Note that both our approach and bread emoji used similar validation strategies with sampling from training data. All models are evaluated on our validation set unless otherwise specified.

B. Pretraining Effectiveness

Table I demonstrates the progressive improvements from our pretraining strategy. Masked-only pretraining reduces validation MSE by 22.3% compared to from-scratch training (44.9k vs. 57.8k). Our multitask approach further improves performance by 10.4% over masked-only pretraining (40.2k vs. 44.9k), confirming that combining reconstruction with policy prediction provides richer representations for difficulty assessment.

TABLE I: Pretraining ablation on validation set

Pretraining Strategy	Validation MSE	Difference
No pretraining (from scratch)	*57.8k*	-
Masked squares only	*44.9k*	-22.3%
Multitask	**40.2k**	**-10.4%**



(a) Masked input board (b) Reconstruction target (c) Policy prediction target

Fig. 1: Multitask pretraining objectives: (a) Input board with randomly masked squares (red); (b) Target state for masked-square reconstruction; (c) Target solution move (arrow) for policy prediction.

C. Validation Set Comparisons

Table II compares GlickFormer v2 against key baselines on our validation set. Our approach reduces MSE by 50.2% compared to GlickFormer v1 and by 13.9% compared to Woodruff et al. [4], demonstrating significant improvements over state-of-the-art approaches.

TABLE II: Validation set comparison with baselines

Method	Validation MSE	Difference
GlickFormer v1	80.8k	-
Woodruff et al. (SotA)	46.7k	-
GlickFormer v2 (ours)	**40.2k**	**-50.2% vs v1** **-13.9% vs SotA**

D. Competition Results and Calibration Impact

GlickFormer v2 achieved third place in the 2025 competition with a test MSE of 55.9k. The top-performing teams secured first and second place with MSEs of 52.3k and 54.4k respectively. To quantify the impact of post-hoc calibration, we evaluated its effect on a preliminary test dataset (subset of official test data). This analysis revealed calibration was critical for performance - without it, our MSE increased by 38.6% to 81.6k, demonstrating its importance for addressing distribution shifts in the full competition environment. The close margin between top 5 solutions (52.3k - 61.1k) highlights the competitiveness of this year's challenge, with our approach placing solidly within this range.

E. Key Findings

Analysis reveals several important insights. Masked-only pretraining provides significant gains over from-scratch training, reducing validation MSE by 22.3%. Our multitask approach yields a further 10.4% improvement over masked-only pretraining, confirming the value of combining reconstruction with policy prediction. Post-hoc calibration proves essential for test performance, preventing a 38.6% MSE degradation. Overall, our solution achieves a 50.2% improvement over our previous entry [1] while reducing training time by >80% through architectural simplifications.

The integration of Maia solve-rate priors contributed substantially to our 50.2% improvement over GlickFormer v1, highlighting the value of human-aligned signals in puzzle difficulty modeling. This demonstrates that encoding domain-specific human behavior patterns through pretrained engines provides crucial complementary information beyond board-state analysis alone.

V. Conclusion

GlickFormer v2 demonstrates that spatial-only transformers with targeted pretraining can effectively predict chess puzzle difficulty. The model's novel multitask pretraining strategy, combining masked-square reconstruction with solution policy prediction, significantly outperforms both training-from-scratch and masked-only pretraining baselines. By incorporating human-centric priors through Maia solve-rate predictions and addressing distribution shifts via post-hoc calibration, the approach maintains effectiveness across varied evaluation environments.

Our solution achieved third place in the FedCSIS 2025 Challenge: Predicting Chess Puzzle Difficulty —- Second Edition, showcasing the potential of efficient architectures combined with strategic pretraining.

For future work, this pretraining framework could be adapted for other chess-specific tasks such as value and policy prediction in chess engines. Additional research directions include enhancing robustness to distribution shifts through domain adaptation techniques and refining temporal modeling approaches for complex multi-move puzzles.

References

[1] S. Miłosz and P. Kapusta, "Predicting chess puzzle difficulty with transformers," in *2024 IEEE International Conference on Big Data (BigData)*, Washington, DC, USA, 2024. doi: 10.1109/BigData62323.2024.10825919 pp. 8377–8384.

[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 *Proceedings of the 20th Conference on Computer Science and Intelligence Systems*, ser. Annals of Computer Science and Information Systems, M. Bolanowski, M. Ganzha, L. Maciaszek, M. Paprzycki, and D. Ślęzak, Eds., vol. 43. Polish Information Processing Society, 2025. doi: 10.15439/2025F5937. [Online]. Available: http://dx.doi.org/10.15439/2025F5937

[3] Z. Tang, D. Jiao, R. McIlroy-Young, J. Kleinberg, S. Sen, and A. Anderson, "Maia-2: A unified model for human-ai alignment in chess," in *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*, 2024. doi: 10.48550/arXiv.2409.20553 Accepted @ NeurIPS 2024.

[4] 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)*, Washington, DC, USA, 2024. doi: 10.1109/BigData62323.2024.10826037 pp. 8415–8422.

[5] 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," in *2024 IEEE International Conference on Big Data (BigData)*, Washington, DC, USA, 2024. doi: 10.1109/BigData62323.2024.10825289 pp. 8423–8429.

[6] S. Bjorkqvist, "Estimating the puzzlingness of chess puzzles," in *2024 IEEE International Conference on Big Data (BigData)*, Washington, DC, USA, 2024. doi: 10.1109/BigData62323.2024.10825991 pp. 8370–8376.

[7] A. Schutt, T. Huber, and E. Andre, "Estimating chess puzzle difficulty without past game records using a human problem-solving inspired neural network architecture," in *2024 IEEE International Conference on Big Data (BigData)*, Washington, DC, USA, 2024. doi: 10.1109/BigData62323.2024.10826087 pp. 8396–8402.

[8] A. Rafaralahy, "Pairwise learning to rank for chess puzzle difficulty prediction," in *2024 IEEE International Conference on Big Data (BigData)*, Washington, DC, USA, 2024. doi: 10.1109/BigData62323.2024.10825356 pp. 8385–8389.

[9] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020. doi: 10.48550/arXiv.2010.11929. [Online]. Available: https://arxiv.org/abs/2010.11929

[11] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," OpenAI, Tech. Rep., 2018. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020.

[13] D. Noever, "Chess transformer: Mastering the game of chess with attention," *arXiv preprint arXiv:2008.04057*, 2020.

[14] D. Misra, "Mish: A self regularized non-monotonic activation function," *arXiv preprint arXiv:1908.08681*, 2019.

[15] D. Monroe and P. Chalmers, "Mastering chess with a transformer model," *arXiv preprint arXiv:2409.12272*, 2024, describes Lc0's transformer architecture and smolgen position encoding. [Online]. Available: https://arxiv.org/html/2409.12272v1

[16] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. Casas, L. Hendricks, J. Welbl, A. Clark *et al.*, "Training compute-optimal large language models," *arXiv preprint arXiv:2203.15556*, 2022.

[17] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *arXiv preprint arXiv:1711.05101*, 2017.

APPENDIX: UNCERTAINTY-MASKING SIDE COMPETITION

After the main leaderboard closed, the organizers announced an additional task focused on uncertainty estimation. Each team was asked to submit a binary mask over the test set marking exactly 10% of instances deemed most problematic. During scoring, the organizers replaced model predictions with the ground-truth values on the masked subset and recomputed the test metric. Let $[P]$ denote the *perfect-mask* score for a given team's final submission (i.e., the best possible score if the team had masked the true worst 10%), and let $[N]$ denote the score obtained using the team's submitted mask. The side competition ranked teams by the ratio $[N]/[P]$ (lower is better).

Evaluation Setting

Per the organizers' rules, the submitted 10% mask was evaluated on *all three of our final submissions*, using the same mask for each. In our case, these were an uncalibrated prediction run, a prediction run calibrated using the post-hoc rescaling parameters of Woodruff et al., and a prediction run calibrated using the post-hoc rescaling parameters used in this paper. The organizers computed the score across these three final submissions according to the side-competition protocol.

Method

Our approach leverages the post-hoc distribution calibration $R(\cdot)$ (Eq. (8)) to construct a simple, model-agnostic uncertainty proxy. For each test puzzle i, we compute a *calibration sensitivity*

$$\delta_i \;=\; \left|\hat{r}_i^{\mathrm{raw}} - R(\hat{r}_i^{\mathrm{raw}})\right|, \tag{9}$$

where $\hat{r}_i^{\mathrm{raw}}$ is the uncalibrated prediction and $R(\hat{r}_i^{\mathrm{raw}})$ is the calibrated one. Intuitively, δ_i measures how much calibration "needs to move" a prediction to account for the rating-distribution shift (caused by limited solving attempts in the test environment). Large δ_i values occur predominantly at the distribution tails (very easy/very hard puzzles), which are known to exhibit higher instability. We rank puzzles by δ_i and mark the top 10% as the uncertainty mask. This design purposely targets cases where calibration most alters the prediction, and was explicitly intended to reduce error on the uncalibrated submission.

Outcome

This calibration-sensitivity mask achieved the lowest $[N]/[P]$ ratio among all participating teams, winning the side competition. Our ratio was:

$$\frac{[N]}{[P]} = 1.426,$$

which gave us **1st place** out of 9 teams that decided to participate in this additional task. For our final submitted mask:

$$[N] \approx 61.49k, \quad [P] \approx 43.12k.$$

Beyond its simplicity and robustness (no additional training or ensembling), the method aligns tightly with our main-system calibration: it explicitly targets those instances where calibration exerts the largest corrective effect, which empirically coincide with the most uncertain test cases under the competition's setting.