Generating conditional chess puzzles

Aatu Selkee et al.

Aalto University

February 13, 2026

What is our plan:

- Follow your paper on chess puzzle generation with the masked diffusion model[1]
- Condition the generation on the themes and the difficulty of the puzzles
- Apply RL for the masked diffusion model
  - Specifically, we plan to use ELBO-based Sequence-level Policy Optimization (ESPO)[2]
- Perhaps make a UI for people to use the model

[1] Xidong Feng et al. *Generating Creative Chess Puzzles*. 2025. arXiv: 2510.23881 [cs.AI]. URL: https://arxiv.org/abs/2510.23881.

[2] Jingyang Ou et al. *Principled RL for Diffusion LLMs Emerges from a Sequence-Level Perspective*. 2025. arXiv: 2512.03759 [cs.CL]. URL: https://arxiv.org/abs/2512.03759.

What have we have done so far:

- Pretrained the masked diffusion model to 300 000 steps (validation loss starting to separate from the train loss slightly, but we may try to train a little further)
- Implemented the RL training pipeline
- Implemented the reward functions (as similarly to your implementations as we could based on the paper)
- Early experiments with the RL training

Reward function components:

- Uniqueness is probably pretty similar to your implementation
- Counter-intuitiveness check might differ
  - We currently have two terms: Stockfish critical point depth and the captured material value. Only the critical depth matters in the current implementation, as the coefficient of the captured material is so low.
- Legality check
- Piece count regularization
- Intra- and inter-batch fen and principal variation distances
- No token-level entropy check currently
- Themes in the generated puzzles match the ones the generation was conditioned on

$$R = \begin{cases} -2 & \text{if not legal} \\ 0 & \text{legal, but no unique solution} \\ 2I_{cnt} + 0.5I_{pieces} + 0.5I_{themes} & \\ \quad + 0.5 \sum_{i \in \{intra, inter\}} \sum_{j \in \{fen, pv\}} I_{i,j} & \text{otherwise} \end{cases}$$

ESPO:

- Mostly the same as GRPO, but with the ELBO instead of the probability of model generation.
- Optimize the following:

$$\mathcal{J}_{\text{seq}}(\pi_\theta) = \mathbb{E}_{x\sim\mathcal{D},y^{(1:G)}\sim\pi_{\text{old}}(\cdot|x)}\left[L_i(\theta)\right],$$

$$L_i(\theta) = \frac{1}{G}\sum_{i=1}^{G}\min(\rho_{\text{seq}}^{(i)}\hat{A}^{(i)}, \text{clip}(\rho_{\text{seq}}^{(i)}, 1-\epsilon, 1+\epsilon)\hat{A}^{(i)}),$$

where $\rho_{\text{seq}}^{(i)} = \exp(\frac{1}{L}(\mathcal{L}_\theta(y^{(i)}|x) - \mathcal{L}_{\theta_{\text{old}}}(y^{(i)}|x)))$ and $\mathcal{L}$ is the evidence lower bound.

- Initial tests with an easy reward have worked well
- If the larger runs do not end up working, we may go back to a more traditional policy gradient.

Results after supervised training:

Table: The proportion of positions satisfying a criterion. An average is taken over 1000 generated positions from the model or 1000 randomly sampled positions from the Lichess Puzzle dataset. The values in the parenthesis are the corresponding values in Feng et al., *Generating Creative Chess Puzzles*.

	Lichess puzzles	Masked Diffusion
Legal	100%	96.8% (99.72%)
Unique	81.4% (95.25%)	9.71% (30.89%)
Counter-intuitive	5.0% (2.25%)	1.1% (1.11%)
Puzzle	4.1% (2.14%)	0.1% (0.34%)

Positions

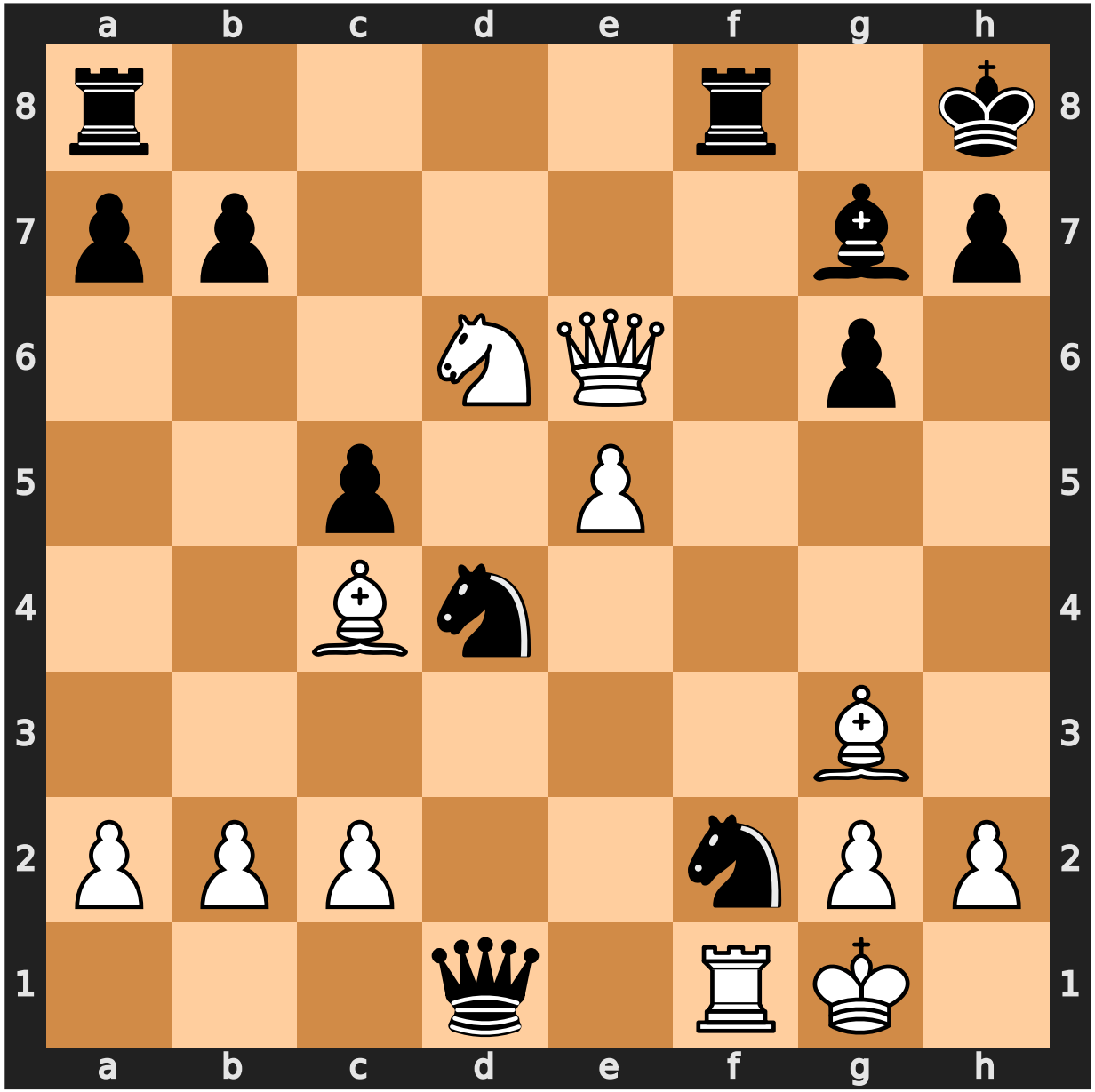

Figure: Themes: middlegame, mate, one move, smothered mate. Rating: 2174

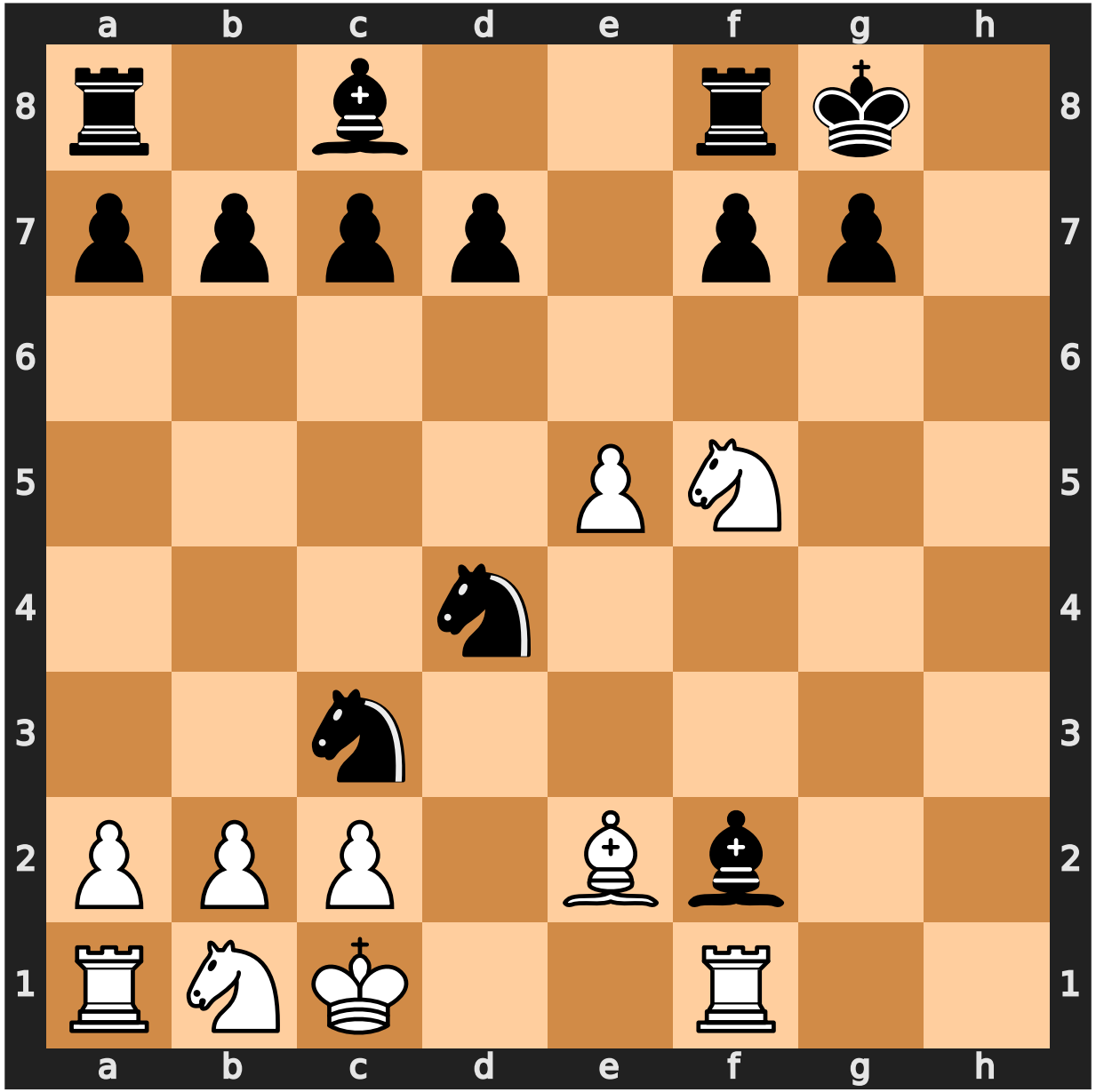

Figure: Themes: long, mate, opening, anastasia mate, mate-in-5. Rating: 1135

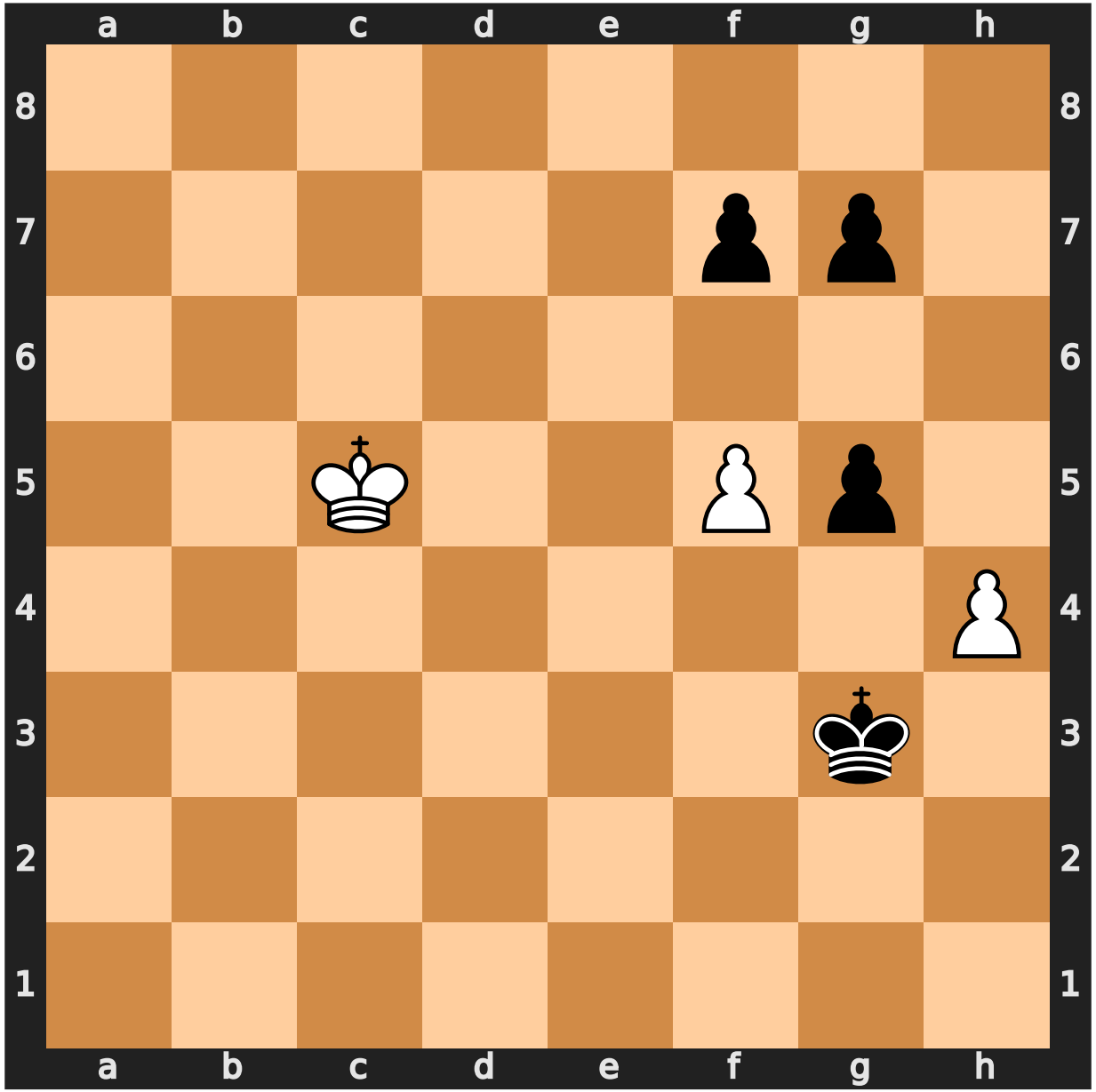

Figure: Themes: crushing, long, endgame, pawnendgame, intermezzo. Rating: 1249