Generating open source chess puzzles - research plan

Aatu Selkee

January 26, 2026

Contents

1 Background

Google DeepMind released a paper about generating novel chess puzzles with generative models at the end of October 2025 [1]. They had a goal of training a generative model to output truly creative and novel outputs. Being a very difficult problem, they decided to start by considering a slightly smaller and easier problem in the field of chess. In this project, we aim to follow the approach by Google in [1] to develop an open-source version of the generative model. We modify the approach in select parts for instance, by conditioning the output on specific themes. In the end, the final users will be able to select which type of puzzle they want to obtain.

2 Objectives

The research goals of this project are:

- Train a generative model that performs on par with the masked diffusion model in [1] in terms of legality, uniqueness, counter intuitiveness and puzzle scores, as well as novelty metrics. Additionally, train a model that generates puzzles that are indistinguishable from human generated ones.
- Use the masked diffusion model for the whole pipeline (supervised and reinforcement learning). Masked diffusion is not used in the RL phase in [1] and only the autoregressive model is used in RL.
- Condition the generation on the theme and the rating of a puzzle.
- Publish the model and perhaps build a website for people to use it.

3 Methods

Similarly to [1], we divide our training in two different stages, supervised training and reinforcement learning (RL). In the supervised stage, we use the Lichess puzzle dataset [2] as the training set, which contains fen-strings, difficulty ratings and theming of different positions. We tokenize the fen-strings using the approach in [3] and the themes by one-hot encoding (multiple themes can potentially be present in one puzzle). In the future, we may consider using a BERT language model to understand the themes as complete sentences, but at first, the themes are not part of sentances. In total there are 66 different themes in the dataset. In addition to the themes, our model will condition on the difficulty rating of the puzzle.

In the supervised phase, we train a masked diffusion model described in [4]. We use similar hyperparameters as the one used in [1], which result in a model with around 200M parameters in total. After supervised training, we evaluate the masked diffusion model with the legality, uniqueness, counter intuitiveness, puzzle and novelty metrics to compare with the final model after RL training.

In the reinforcement learning phase, we follow exactly the same procedure as in [1], but use the masked diffusion model instead of the autoregressive model. In [1], the authors chose the autoregressive model based on the supervised phase and only used it for the RL. We use critic-free PPO as the optimization method and the same reward structure as in [1]. We may also experiment with additional rewards given based on the generated puzzle matching the theme it was conditioned on.

After reinforcement learning, we compare the legality, uniqueness, counter intuitiveness, puzzle and novelty metrics with the ones before RL as well as the ones in the paper. In addition, we will perform a human study, where we show random puzzles from the Lichess puzzle dataset [2] and the ones we generated to human experts. We will let them rank the puzzles in realism, difficulty, creativity, fun and counter intuitiveness in scale from 0 (poor) to 3 (perfect) as in [1]. These metrics can be compared with the ones presented in [1]. Additionally, we will let the experts guess if the puzzle was generated by our model or not, and compare if the theme and the difficulty rating match the puzzle itself.

4 Timeline

As the authors of [1] had trouble with the stability of the RL algorithm, we expect the RL phase to take the most amount of time. Hence we allocate more time for the RL than for the supervised phase. The timeline will look something like the following:

- January:
  - All practicalities
  - Experiment with the dataset
  - Implement dataloaders and split the test set
  - Implement the tokenization
  - Implement the model
  - Start working on the supervised phase
- February:
  - Supervised model finished
  - Generate puzzles with the supervised model
  - Compute the chosen metrics
  - Start working on the RL phase
- March:
  - Continue working on the RL
  - Contact chess experts for the experiment
- April:
  - Finish the RL training
  - Generate new puzzles and compute the metrics
  - Contact chess experts for the experiment
- May:

  - Do the experiment with the experts
  - Start writing the paper and the thesis

- June:
  - Finish the thesis and send the publication for review
  - Build a website where people can use the model or publish the model on Huggingface or similar

5 Results

The preliminary results after supervised training the masked diffusion model are shown in the table below.

Table 1: 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 [1].

	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%)

References

[1] X. Feng, V. Veeriah, M. Chiam, M. Dennis, R. Pachauri, T. Tumiel, F. Barbero, J. Obando-Ceron, J. Shi, S. Singh, S. Hou, N. Tomašev, and T. Zahavy, “Generating creative chess puzzles,” 2025.

[2] lichess.org, “Lichess puzzle dataset,” 2025.

[3] A. Ruoss, G. Delétang, S. Medapati, J. Grau-Moya, L. K. Wenliang, E. Catt, J. Reid, C. A. Lewis, J. Veness, and T. Genewein, “Amortized planning with large-scale transformers: A case study on chess,” 2024.

[4] J. Shi, K. Han, Z. Wang, A. Doucet, and M. K. Titsias, “Simplified and generalized masked diffusion for discrete data,” 2025.