PAIRWISE LEARNING TO RANK FOR CHESS PUZZLE DIFFICULTY PREDICTION

COMPETITION

- Dataset
  - ~4M of puzzle instances
  - Data: starting position, moves, rating, rating deviation, tags, etc.
- Goal: estimate puzzle rating
- Evaluation metric: Mean Squared Error (MSE)

PUZZLE RATING CALCULATION ON LICHESS

- Glicko-2 rating system to rate players and puzzle
- Components:
  - Rating r: Represents the skill level or difficulty of a player or puzzle
  - Rating Deviation RD: Measures the uncertainty in the rating
  - Volatility σ: The degree of expected fluctuation in the rating
- Scaled components:
  - $\mu = \frac{r - 1500}{173.7178}, \phi = \frac{RD}{173.7178}$

PUZZLE RATING CALCULATION ON LICHESS

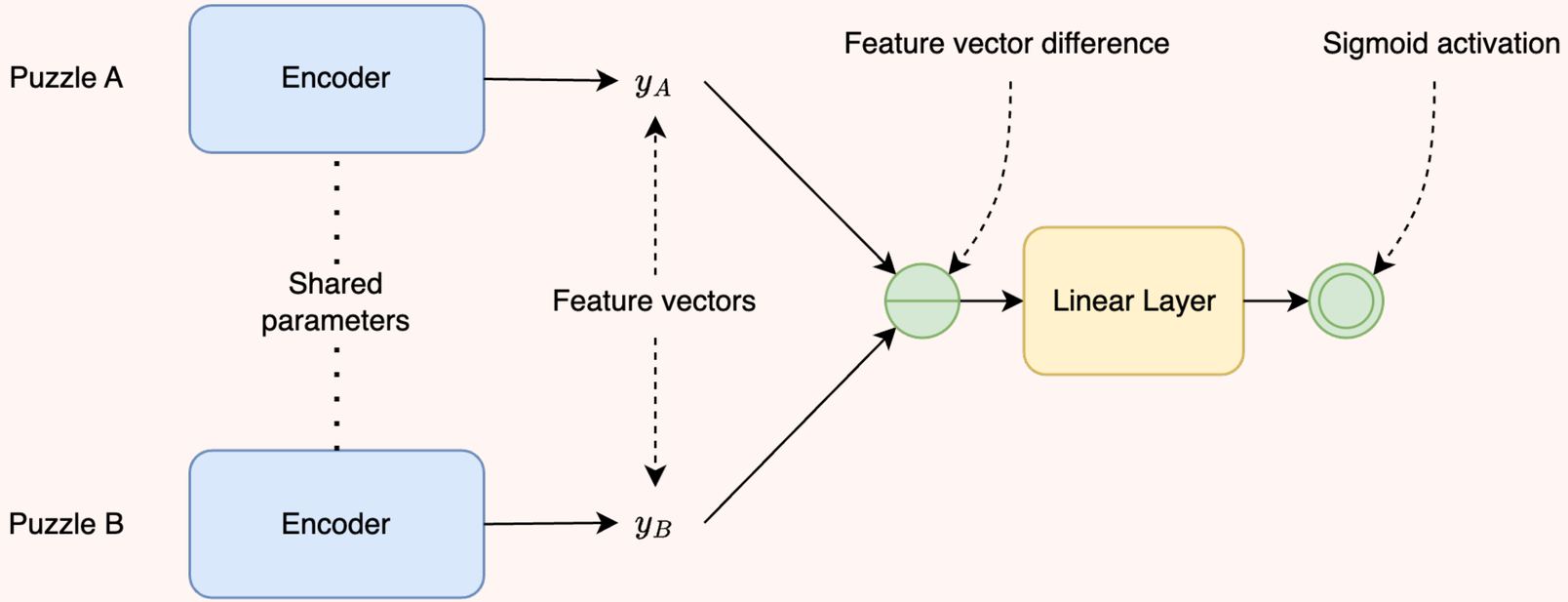
- **Players and puzzles are assigned initial values:**
  - $r = 1500, RD = 350, \sigma = 0.09$
- **Player attempts to solve puzzle**
- **Player and puzzle r, RD and σ are updated accordingly**

APPROACH

- Train a model that can compare relative difficulty between pairs of puzzles
- To estimate the rating of a puzzle:
  - Simulate games between this puzzle and puzzles from the training set
  - Compute the estimated the Glicko-2 rating

MODEL ARCHITECTURE

- **Based on RankNet**
- **Encoder: CNN or Vision Transformer**



INPUT REPRESENTATION

- 8x8xC binary image
- Position planes
  - Binary planes representing the pieces present on the chessboard
- Legal moves planes
  - Castling rights planes
  - Move/capture/checks planes
- 12 positions per puzzle -> 52 planes per puzzle

TRAINING

- **Sample pairs of puzzles from the training set**
- **Use Binary Cross Entropy loss**
- **Target probabilities derived from Glicko-2 expected outcome formula**
  - $p = \dfrac{1}{1 + \exp\left(-g\left(\phi_{AB}\right)\left(\mu_A - \mu_B\right)\right)}$
  - **With** $\phi_{AB} = \sqrt{\phi_A + \phi_B}$**, and** $g(\phi) = \dfrac{1}{\sqrt{1 + 3\phi^2/\pi^2}}$

INFERENCE

- Initialize puzzle rating, rating deviation, and volatility
- Sample K reference puzzles from the training set
- Simulate pairwise comparison using the trained model
- Use the Glicko-2 algorithm to compute the final rating

RESULTS

- Comparaison between ResNet and Vision Tranformer backbone
  - CNN and Vision Transformer had similar performance in regression setting
  - Vision Transformer outperformed CNN in the LTR setting
- Final results on the private test set: 129245.2292 MSE, 4th place

RESULTS ON THE PUBLIC TEST SET

Model	MSE
Vision Transformer LTR	61381.3812
ResNet LTR	68632.2044
Vision Transformer Regression	77103.7790
ResNet Regression	76651.6022

CONCLUSION

- Key outcomes:
  - The Transformer-based model delivered best results compared to the CNN-based one
  - Ranking-based models demonstrated stronger performance compared to regression-based ones
- Ideas for future work:
  - Integrate additional features (puzzle tags/themes, handcrafted features, engine-derived features, etc.)
  - Leverage pretrained chess models

THANK YOU!