Project Presentation: Improving RBC agent

Shubh Kumar
Puranjay Datta

30-01-2023

Outline

What is RBC ?

- Reconnaissance Blind Chess (RBC) is a chess variant which unlike chess is an imperfect information game.
- Every turn comprises of two parts Sense(3×3 board) followed by Move.
- An illegal move will result in no action being taken.
- The competition is a 15 min+5 sec increment with a 50 move draw rule.
- This uncertainty makes decision making harder and we have to come up with clever mixed strategy policies.
- In the RBC Competition for NeurIPS 2021 Fianchetto came first and in 2022 came second(defeated by Strangefish2).
- Sample Game

How is fianchetto modelled ?

- Modelled RBC using a POMDP with belief updated at each move using Bayes' rule and Opponents move policy.
- Opponents move $Pr(a|s)$ estimated using LC0 and RBC specific heuristics.
- $Pr(s'|b,z) = \sum_a Pr(s'|b,a,z) \sum_s Pr(a|s)Pr(s|b)$
  $= \sum_a Pr(z|s',a) \sum_s Pr(s'|s,a)Pr(s/b) \sum_s Pr(a|s)Pr(s|b)$
- $Pr(a|s) = softmax(Lc0LastLayer(s) + c)$

  c :incentive vector

  a :opponent's move

  s' :new state(after opponent's move)

  s :old state(before opponent's move)

  b :belief(probability) over the old states

  z :observation

Objectives

1. Implement the Replay buffer
2. Analyze the blunders using it
3. Improve the opponent modelling
4. Addressing other concerns

Replay buffer

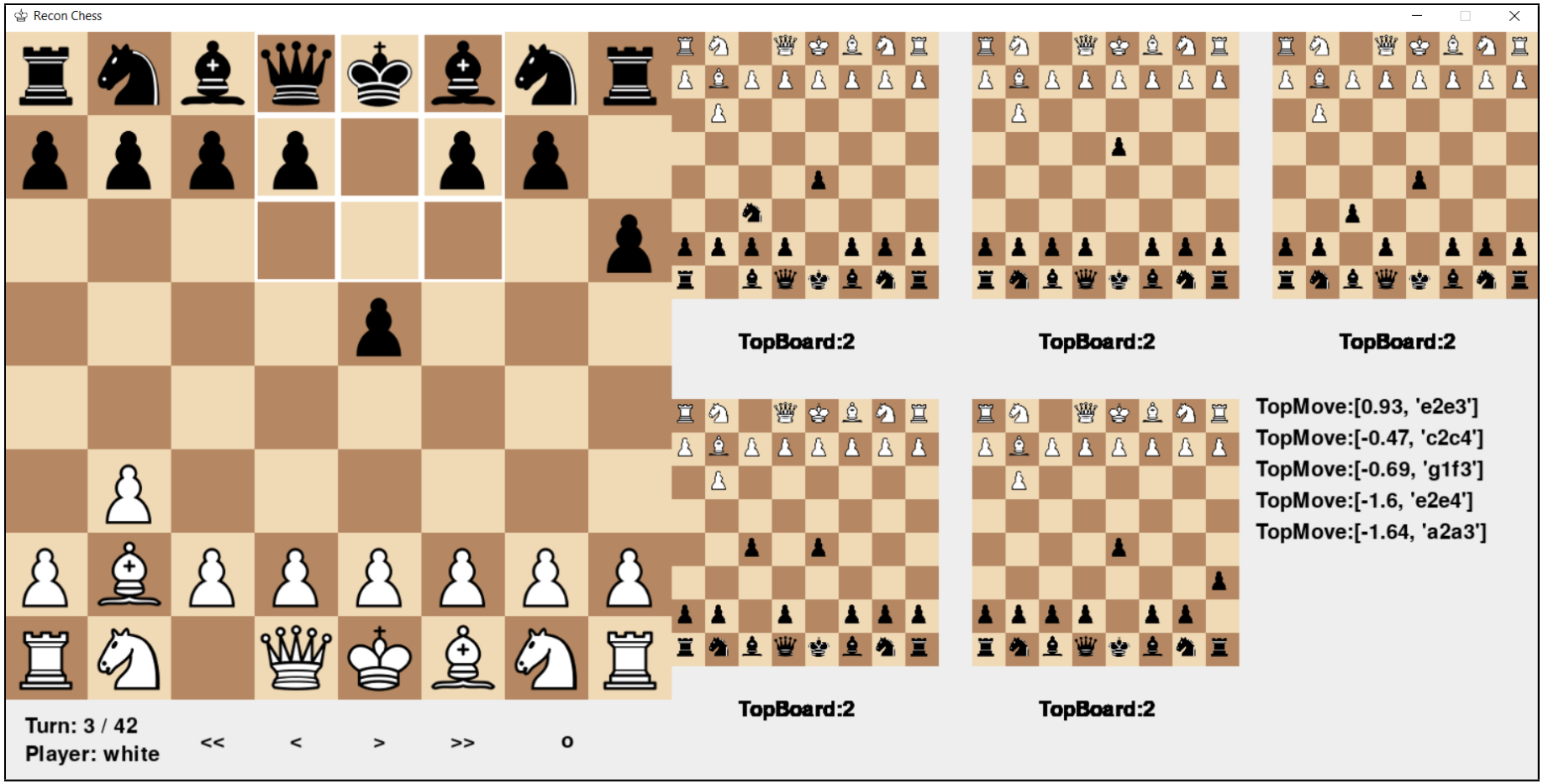

Figure: Replay Buffer GUI

Blunder 1

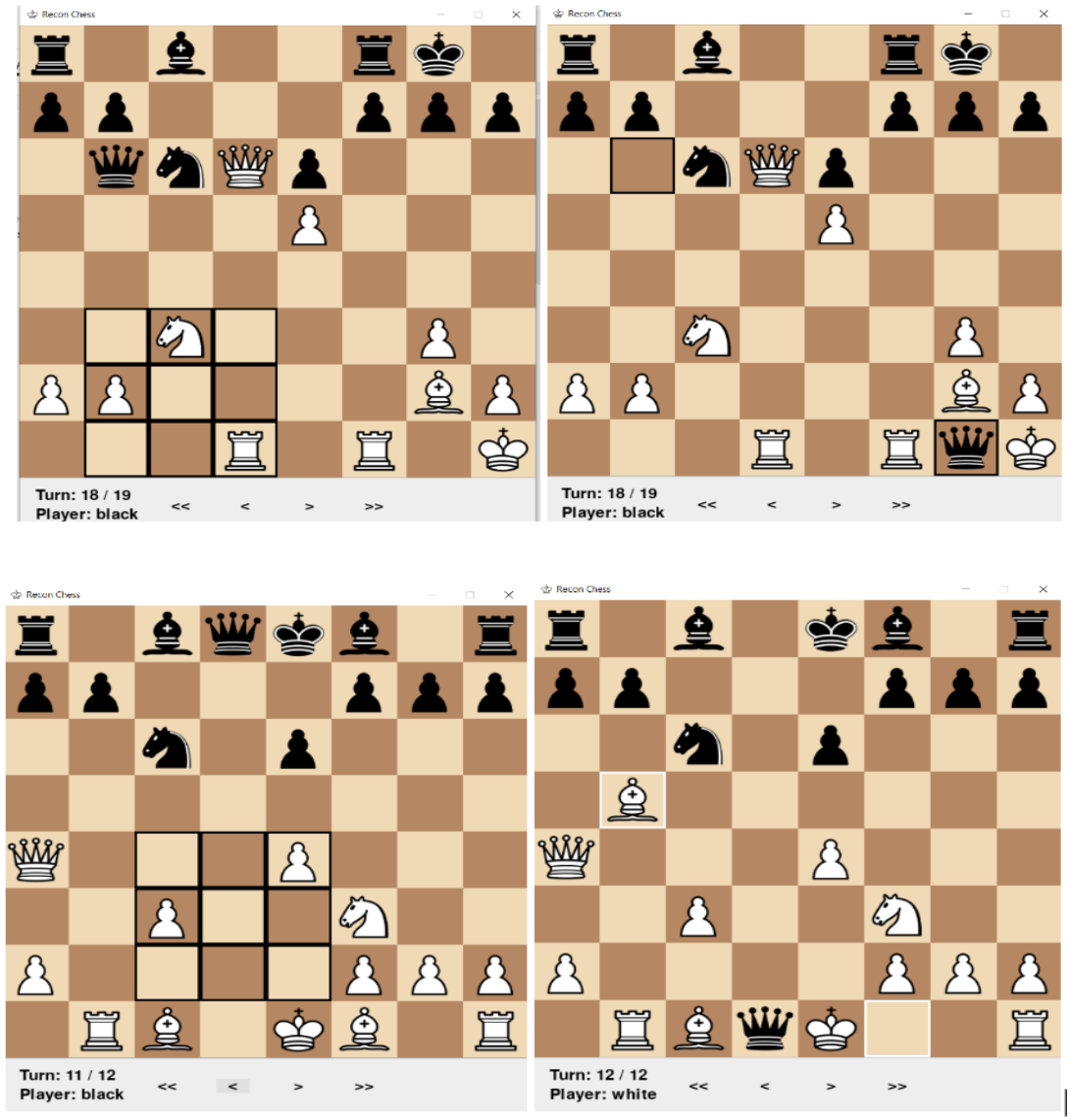

Figure: Sneak Attack

Blunder 2

Figure: a) Not a poisoned pawn-b) Forgot to take the Queen-c)Why not h5

Blunder 3

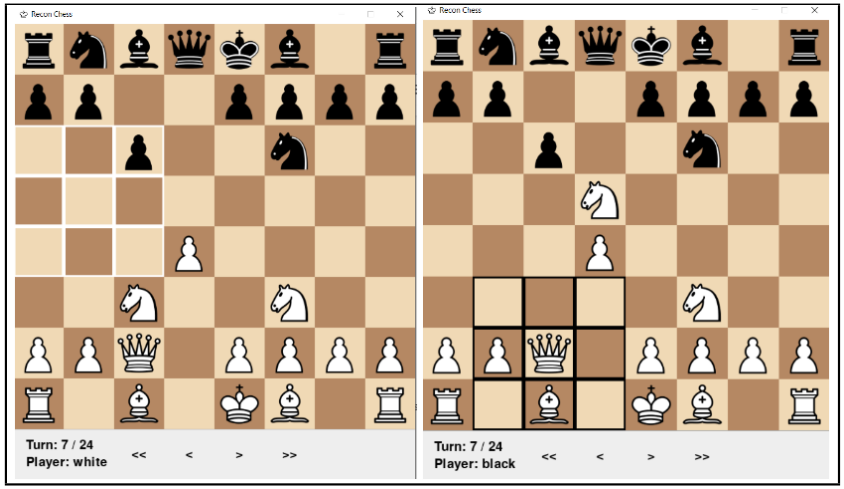

Figure: Sacrificed our knight for no compensation

Figure: Top board in the set with probability=0.309

Pre-conceived Improvements

Work has been done towards improving Fianchetto in the past, which remains to be integrated, besides addressing concerns that the original creators had.
We shall aim to address some of these.

Improving Opponent Modelling

- Currently, Fianchetto evaluates best moves for the various possibilities for the board using LeelaChess Zero's pre-trained Neural Models.
- But work done by Gurnoor and Dhruv as an RnD Project over past semesters have had ≈ 12 % improvement over LeelaChess's Model.
- Their Model takes a Transfer Learning Approach, to achieve this feat.
- They trained and validated their model on histories of RBC games that have been played over time on their servers.
- We shall try to replace LeelaChess with their model.

Addressing Past Concerns

There were several concerns that the people who designed Fianchetto in the first place had in mind. These included

- Uniform Board Sampling
- Weighing Actions

Uniform Board Sampling

- At present, While the game moves on, We maintain a set of all possible boards, but do not associate any probability distribution with them.
- Before every turn, We feed these boards to LeelaChess, get the optimal actions (which are softmax(ed)), average the probabilities for each action out, and play the optimal move.
- But owing to the large number of moves, and relatively small number of possible boards, this leads to a pretty random utilization of the information at hand.
- We shall aim to get this probability distribution by maintaining a game tree.
- We know the position of all our pieces, the opponent's pieces are at unknown locations, but we may approximate their moves for a particular board using LC0, and then use the action probabilities from LC0 as probabilities for moves that the opponent might have played.
- We would then purge this tree based on our sensing.

Weighing Actions

- As of now Fianchetto feeds boards from its sample set to LC0, gets the weight for each move, then averages to determine the final move, but this approach doesn't make sense.
- Suppose we define a utility function $u : boards \rightarrow \mathcal{R}$, which is the utility of the board for a player provided its their own turn.
  The function we optimize should therefore be :

$$a_{optimal} = \arg min_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}} P(b) u(T(b, a)) \tag{1}$$

where $\mathcal{A}$ is the set of all possible moves.
$\mathcal{B}$ is our sample set of boards
$P(b)$ is the probability of the current board setting being b
$T(b, a)$ denotes the board we'll reach if we play move a on board b

Weighing Actions

- What Fianchetto is instead doing is somewhat shady, Its optimizing :

$$a_{optimal} = \arg max_{a \in \mathcal{A}} \sum_{b \in \mathcal{B}, a_{optimal}(b)=a} P(b) \frac{e^{-u(T(b,a))}}{\sum_{a \in \mathcal{A}} e^{-u(T(b,a))}} \quad (2)$$

- Besides having an awkward way to get $P(b)$, i.e. By giving higher weight to boards which have a higher utility function for the opponent.

Baselines

- Fianchetto was defeated by StrangeFish2 last year.
- RBC's official website actually allows us to play our bot against StrangeFish2 alongside a few classic bots which either keep track of a number of board states or make use of Monte Carlo Regret Minimization.
- We shall play our bot against these a number of times after each improvement, to see how much of an improvement we have over the past versions of Fianchetto.
- Test new model at some crucial junctures where the bot was making a blunder and observe its new move & board sets with the help of replay buffer we would have created and also compare based on accuracy of the opponent move predicted.

Conclusion

- Fianchetto's skeleton has been largely adapted from StrangeFish (the predeccessor of StrageFish2) which was made public in 2020.
- One of Fianchetto's major strengths is its use of prediction capabilities of LeelaChess, which differentiates it from StrangeFish.
- We hope to see improvements in its game as we integrate better prediction and fix more of the basic issues which ailed it.

Thank You!