SARFA+: Extending Explainability in Chess

Rishi Raman, Rohan Mehta, Arsh Singhal, Rithesh Rajasekar

***Abstract*—Model explainability has become a crucial area of research for Deep Reinforcment Learning (DRL) to build a culture of safety and transparency. Our work is inspired by SARFA and provides new methods for perturbation-based explainability. We first extend the SARFA framework to evaluate the importance of absent features and conceptualize offensive versus defense saliencies for adversarial games. Next, we create a temporal version of SARFA called Sequential SARFA to understand feature importance in the context of longer term strategies. Lastly, we develop a novel framework called Pairwise Importance for RL Sensitivity, PaIRS, which is the first method to our knowledge that identifies important groups of features for decision making. Our methods are evaluated both qualitatively and quantitatively within the game of chess. Our work serves as a key contribution to inspire future directions of research in explainable DRL.**
Github Code Repository Link
Video Presentation Link

I. Introduction

As modern deep learning approaches rely on larger and larger models, it is increasingly difficult to understand the reasoning behind a model's output. However, the ability to explain a model's decisions has grown to become more important as models are being deployed into more sensitive and mission-critical environments. For example, a model that is responsible for hiring decisions must have a sufficient explainability to ensure fairness and avoid social biases.

Recent works in the domain of Explainable AI (XAI) have primarily focused on transparency within the language domain, specifically Large Language models [1], and vision domain, specifically Vision Transformers [2]. Deep Reinforcement Learning (DRL) models have traditionally struggled with real-world success due to the massive datasets and compute time required. With modern advances in both storage systems and GPUs, DRL models are more practical for real-world use cases in several domains including robotics, language, healthcare, and adversarial agents. Thus, DRL model explainability is an important frontier of research.

There are already multiple approaches to explain DNN models. Several state-of-the-art methods, including Grad-CAM, rely on the gradient of the model [3]. This inherently assumes access to the underlying model itself, which is often not the case during a model's deployment life cycle. Our novel approaches focus on model explainability in limited information environments where only the model's outputs are accessible. Our works are inspired by SARFA which uses feature perturbations to create a saliency map using only q-values, highlighting the relative importance of a feature in a state for a DRL model's action selection process [4].

In this paper, we use a standard chess environment to evaluate the applicability of our explainability strategies. Chess is a particularly interesting and useful environment for exploring explainability for a few key reasons. There exist incredibly strong chess engines capable of giving a single numerical evaluation of the position along with the best move(s) in the position. The notion of a numerical evaluation is directly analogous to a Q-value, allowing us to focus on gathering useful insights from only the engine evaluations rather than the mechanics of accurately evaluating positions. This will allow our proposed techniques to easily be generalized to other environments and models with a trustworthy Q-function. Chess is also a rich enough problem where everything from simple, surface-level insights to deep multi-move plans can be targets for explanation. In addition, our methods can also have immediate utility to chess players of different strengths.

In this paper, we introduce multiple novel approaches for explainable DRL. Below, we have listed the primary contributions of our work:

(1) Extend the SARFA framework to evaluate the importance of feature absences through additive perturbations.
(2) Extend the SARFA framework to identify offensive versus defensive feature saliencies in adversarial environments.
(3) Introduced a novel technique called Sequential SARFA, which uses temporal perturbations on top of naive SARFA, to capture temporal strategies.
(4) Created a novel framework called Pairwise Importance for RL Sensitivity or PaIRS for highlighting important groups of features to explain RL decision-making and potential strategies.

II. Related Works

Model explainability has recently become a more prominent area of research, with many works extending from the language and vision domains for DRL applications.

A. Feature Attributed-Explainability

One of the most successful areas of explainability is feature attributed-based explanations which measures the relevance of each input feature for a model's prediction [1]. The first method is pertubation-based which alters the input examples by removing or modifying features and evaluates the change in outputs to derive conclusions. In the domain of natural language, perturbations have been introduced in various levels of the information hierarchy including the hidden units, words, and the rudimentary tokens itself [5], [6], [7]. The second method popular method includes gradient-based explanations which evaluates the partial derivative of the output with respect to each input feature to determine the feature's importance.

This has shown utility in the language domain with token-level attribution scores [8] and in the image domain with a technique called GradCAM [3].

B. Attention-Based Explanations

Another promising avenue of research is attention-based explanations which attempt to explain a model's output based on identifying the parts of the input are most attended to [1]. This has been demonstrated by visualizing self-attention with LLMs [9].

For our work, we focus on perturbation-based methods since this does not require access to the underlying model or attention-based mechanisms, extending well to a majority of RL frameworks. Thus, our work has drawn inspiration from SARFA which is one of the leading perturbation-based explainability metrics for RL [4].

C. SARFA Metric

SARFA is a perturbation-based saliency approach, where for a given action, we perturb s to get s' for all features and compare $Q(s', a)$ to $Q(s, a)$ and compute a saliency score for each feature for a given action [4]. There is the possibility that the legal action may not be possible after a perturbation, so we only utilize the actions allowed in both states, $\mathcal{A}_s \cap \mathcal{A}_{s'}$.

There are many ways to generate a saliency measure from these Q-values, but SARFA's saliency measure has two key properties:

Specificity: SARFA wants the perturbation to a salient feature to affect a specific action more than any other actions, meaning that feature is particularly important to that action [4]. More concretely, for a feature f to be specific to an action $\hat{a}$, we want $Q(s, \hat{a}) - Q(s', \hat{a}) >> Q(s, a) - Q(s', a)$ for all $a \neq \hat{a}$.

SARFA's measure of specificity is defined as $\Delta p = P(s, \hat{a}) - P(s', \hat{a})$ where

$$P(s, \hat{a}) = \frac{\exp(Q(s, \hat{a}))}{\sum_a \exp(Q(s, a))}$$

Relevance: It is possible that a state s' has substantially different Q-values for all actions compared to s, however, SARFA only cares about perturbations that are relevant to the action at hand and not the state in general [4]. SARFA's measure of relevance is calculated as the KL-Divergence $D_{KL} = P_{\text{rem}}(s', a) || P_{\text{rem}}(s, a)$ where

$$P_{\text{rem}}(s, a) = \frac{\exp(Q(s, a))}{\sum_{a' \neq \hat{a}} \exp(Q(s, a'))} \quad \forall a \neq \hat{a}$$

The SARFA saliency metric is calculated as [4]:

$$S[f] = \frac{2K\Delta p}{K + \Delta p}$$

where $K = \frac{1}{1+D_{KL}}$.

Since $0 \leq K \leq 1$ and $0 \leq \Delta p \leq 1$, $0 \leq S[f] \leq 1$.

The SARFA paper includes many examples of the robustness of this metrics and focuses on understanding which features are salient to an action. This is accomplished by removing a feature when perturbing a state.

In this paper, we will be using the notation $S_a[f]$ to show that the saliency of a state-feature is calculated with respect to a given action a.

III. Method

We are given an environment with a state space $\mathcal{S}$, an action space $\mathcal{A}(s)$ for $s \in \mathcal{S}$ and a Q-function $Q(s, a)$ for $s \in \mathcal{S}, a \in \mathcal{A}(s)$. We will assume that an agent follows a greedy policy where at a given state s, the agent's chosen action $\hat{a} = \arg\max_a(Q(s, a))$.

As in the SARFA paper, a given state is parameterized by state features $\mathcal{F}$. For chess, the features are the 64 squares.

Our goal of explaining an agent's actions will be achieved through determining which state features are most salient to the action taken. A high saliency for a given state feature means that that state feature was highly influential in the agent's decision to pick its chosen action. Saliency can be measured through a saliency score $\mathcal{S}$, where $\mathcal{S}[f] \in (0, 1)$.

A. Additive Perturbation

Firstly, intuitively, it is important for certain moves that a set of squares be empty. That is, the *absence* of the feature is important. Our method tests for this by using the SARFA method of perturbation. However, instead of removing a piece, we add a pawn of the current player's color to modify the board so that the feature (square) is no longer absent.

The generalized formula for the saliency with additive perturbations to an empty feature e of a given state s with respect to an action a is as follows:

$$S_a[e] = \max_{p \in P_{s,e}} S_a[p]$$

where P is the set of all legal perturbations to the empty feature. By taking the maximum saliency across all perturbations, it must be the case that if $S_a[e]$ is high, in at least one possible perturbation, then it is crucial for that feature to remain empty, meaning that the empty feature is salient.

In chess, a good simplification for calculating the additive saliency is by simply adding a pawn of the same color to the empty square. This causes the least change in the dynamics of the board, so this most accurately evaluates the saliency of the empty square. In environments where P is extremely large, heuristics or even a random sampling can be used to simplify this computation as iterating through all possible perturbations might not be computationally feasible.

B. Offensive and Defensive Saliency

As the next extension, we introduce the concept of offensively and defensively salient pieces. Currently, SARFA is looking at whether perturbing a square or piece specifically makes a move less promising. If so, we say that this square is salient to the move. However, what if perturbing a square specifically makes a move much more likely to occur? This means that that state-feature is performing a defensive role against the move and thus, it is defensively salient.

To figure out which moves are offensively and defensively salient, we can simply look at the sign of Δp in the SARFA equation. If $\Delta p > 0$, we know that the move was significantly less promising *after* the perturbation, thus the state-feature is *offensively* salient to the move. Similarly, if $\Delta p < 0$, we know that the move is less promising *before* the perturbation and thus, the state-feature is playing a *defensive* role against the move.

We note that the two techniques of additive perturbation and offensive/defensive identification may be combined to determine whether an empty square is playing an offensive or defensive role.

C. Sequential SARFA

SARFA attempts to encode sequential data through the Q-values. A better mechanism for tracking temporal saliency information is Sequential SARFA:

Traditional SARFA simply looks at the given position and computes saliency values. However, what if we consider multiple time steps in the environment? This will give us a more representative picture of the saliencies of state-features for future actions.

Our algorithm involves computing all the saliency values d timesteps steps into the future, and combining these values through a discounted summation (similar to a Bellman backup) to prioritize features that play an important role closer to the current time step.

Algorithm 1 Sequential SARFA

Require: State s, Discount Factor γ, Depth d
  $saliencies \leftarrow \{\}$
  $feature\_map \leftarrow \{\}$
  for $feature \in \mathcal{F}$ **do**
    $f\_map[feature] \leftarrow feature$
  end for
  for $step \leftarrow 0$ to $d-1$ **do**
    $opt\_a \leftarrow$ optimalMove(s)
    for all $(pert\_s, pert\_f) \in \mathcal{F}$ **do**
      $saliency \leftarrow$ getSaliency($pert\_s$)
      $saliencies[pert\_s] \mathrel{+}= saliency \cdot \gamma^{step}$
    end for
    $s \leftarrow s$.performAction(opt_a)
  end for
  return $saliencies$

When the algorithm makes a move, we map the new position of the moved piece back to the old position to correctly track saliencies across a moved piece or feature.

There are two algorithmic hyperparameters used in the algorithm. The depth specifies how many actions and board states into the future are considered. The model simulates both friendly actions and enemy actions (a depth of two would have white move and black move for instance). The discount factor is used to decrease the saliency of future important positions when viewing the current state. This is a similar mechanism to Bellman equation [10].

D. PaIRS

Algorithm 2 PaIRS

Require: State $s, compare\_method$, percentile k
  $pw\_sensitivity \leftarrow \{\}$
  $all\_features \leftarrow s$.getFeatures()
  for all $(f_1, s_1^{'}) \in all\_features$ **do**
    for all $(f_2, s_2^{'}) \in all\_features$ **do**
      $sensitivity \leftarrow 0$
      $legal_moves_1 \leftarrow$getMovesPostPerturb($s_1^{'}$)
      $legal_moves_2 \leftarrow$getMovesPostPerturb($s_2^{'}$)
      $action\_set \leftarrow legal\_moves\_1 \cap legal\_moves\_2$
      if $compare\_method = KL$ **then**
        $q_1 \leftarrow s_1^{'}$.getQVals[$action\_set$]
        $q_2 \leftarrow s_2^{'}$.getQVals[$action\_set$]
        $sensitivity =$ entropy(q_1, q_2)
      else
        for all $a \in action\_set$ **do**
          $q_1 \leftarrow s_1^{'}$.getQVals(a)
          $q_2 \leftarrow s_2^{'}$.getQVals(a)
          $sensitivity \mathrel{+}= |q_1 - q_2|$
        end for
      end if
      $pw\_sensitivity[(f_1, f_2)] \leftarrow sensitivity$
    end for
  end for
  $import\_pairs \leftarrow$ bottom k percentile of $pw\_sensitivity$
  Graph $G \leftarrow$ graph using $import\_pairs$
  $import\_groups \leftarrow$ connectedComponents(G)
  return $import\_groups$

So far, our methods have considered how a state-feature is salient to a given action. This new framework that we present attempts to understand how different features relate to each other to explain a DRL model's reasoning.

In PaIRS, the notion of related features is captured by comparing the Q-value distribution of legal actions after perturbations of any two features. If the change in the Q-value distribution is similar between two perturbations, we can say that the two features are related.

There are two methods we use to compare the Q-value distributions. In one, we simply compute the difference between the Q-value deltas caused by the two perturbations. In the other method, we can take the entropy or KL divergence between the two distributions and use it as our sensitivity measure. Both methods result in pairwise sensitivity values between every pair of features, where a lower value indicates more correlation and important relationship between the features.

The bottom k percentile of the pairwise sensitivity values is used as a threshold to define the most important relationships. This is represented as a graph structure, where the nodes are the features. We run DFS on this graph to find the connected components. Each connected component represents an important group of features for the current state.

IV. Results

The experimental results for the methods mainly consist of qualitative findings that highlight the successes and shortcomings of each method on relevant chess boards. In addition to the qualitative results, the Sequential SARFA method was also tested against the saliency dataset used in the original SARFA paper [4]. Although the experiments in this paper focus exclusively on chess, these methods can be extended to many RL environments like other strategy games.

A. Additive Perturbation

The additive perturbation was tested on multiple boards to evaluate its performance qualitatively. In the first board, we can see that Rh7 will deliver checkmate to the black king. However, notice how the rook on a2 along with the bishop on g3 are essential in cutting off squares for the black king to escape to. In addition, it is important that the squares between h7 and the black king on b7 stay open, allowing the white rook to deliver check. Thus, it is important that those pieces don't get blocked since Rh7 would no longer be checkmate and black could deliver checkmate with Rc1 instead. As we can see, our method highlights many of these squares clearly, demonstrating the offensive importance of those empty squares clearly. In the second board, we can see that it is important that d5 and e4 remain empty in order for Bc6 to be a check and win material against the white king. However, there are some shortcomings. Not all of the squares expected are highlighted. This is due to the fact that adding a white pawn may still keep the original optimal move to be the strongest move due to a deeper tactical reason.

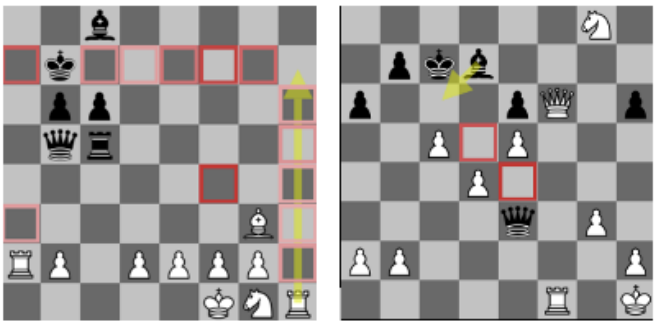


Fig. 1: Additive Perturbation Examples

B. Offensive and Defensive Saliency

We demonstrated offensive and defensive saliency on multiple boards. Specifically, we evaluate the method on two key examples. In the first example, we can see that the bishop on f2 can capture the queen on a7, giving white a decisive advantage. However, due the rooks on d2 and h3 along with the pawns on f7 and g7, white can move their rook from d3 to d8, delivering checkmate. Thus, due to these pieces, Rd8 becomes a more attractive move than Bf2, making them very offensively salient.

The second example demonstrates defensive saliency. White is threatening to move their rook to d8, delivering checkmate. However, the black knight on b7 prevents this move from being effective. Thus, the knight is highlighted in blue as defensively salient. Additional board analyses can be found in Figure 7 (appendix) which showcases the ability of the method to pick up defensive saliency in short and longer horizons.

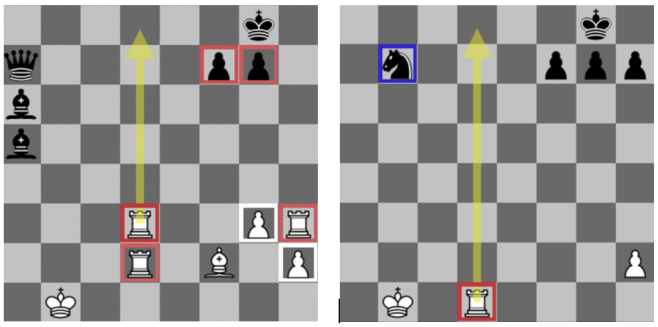


Fig. 2: Offensive and Defensive Examples

C. Sequential SARFA

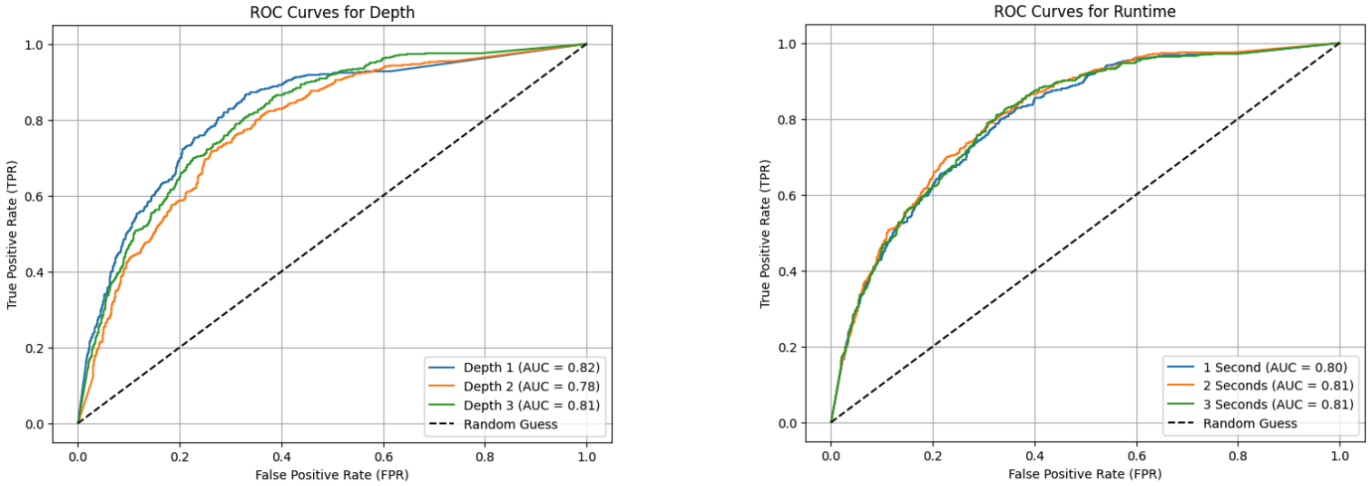
Our primary quantitative analysis was to compare Sequential SARFA to SARFA and the Chess Saliency Dataset published by [4]. The dataset contains 104 unique chess puzzles which with labels for correct moves and saliency as determined by 3 human experts (ELO $>$ 2200). Our implementation of SARFA achieved in accuracy of 79.93% compared to the paper's accuracy of 72.41% due to difference in Q-value model and engine runtime.

Although the authors claim each puzzle has a single correct move, we found that some have a variety of strong sequences of actions an expert could play. In Figure 4, you can see our agent chose a different series of actions compared to the action sequence specified by the dataset. This is most likely due to computational limitations and difference in Q-value models.

Although we used the Chess Saliency Dataset [4] as our primary quantitative comparison, we do not believe it accurately tracks the problem Sequential SARFA solves. SARFA is used to justify the next move taken while Sequential SARFA considers longer term strategies associated with the next move. Although this dataset is not ideal for the problem, it is readily accessible and can approximate the results.

Ablation studies were used to analyze the performance given varying depths and Stockfish runtime. We can see in Figure 3a that depth has a strong influence on performance. As mentioned before, the dataset is poor at measuring the power of Sequential SARFA but provides some evidence of the impact. Additionally, we can see runtime has almost no impact on the performance in Figure 3b. This implies that Q-value do not accurately incorporate next-move saliencies, emphasizing the need for Sequential SARFA.

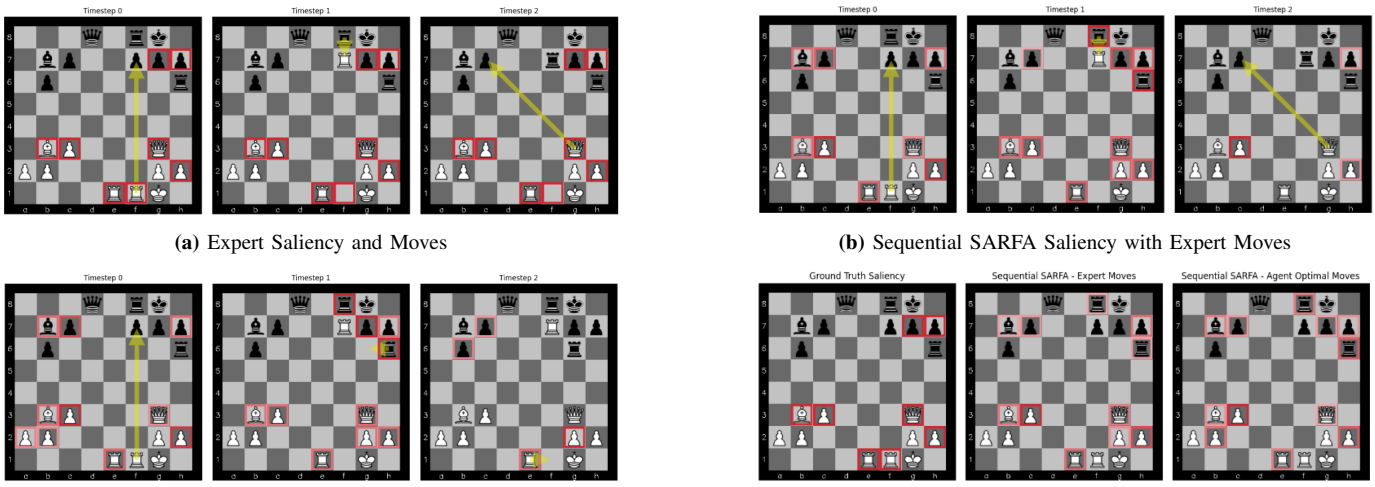
A qualitative analysis of the value of Sequential SARFA (depth of 5) can be seen in Figure 5. The upper timeline in Figure 5a shows the saliency map at each timestep. These figures demonstrate how important rolling out the saliency calculation captures information Q-values cannot. E4 and F7



(a) Sequential SARFA with Different Depths

(b) Sequential SARFA with Different Stockfish Runtimes

Fig. 3: The above figures describe the performance of different



(a) Expert Saliency and Moves

(b) Sequential SARFA Saliency with Expert Moves

(c) Sequential SARFA Saliency with Optimal Moves

(d) Final Saliency Maps Comparison

Fig. 4: Comparing saliency maps between dataset results (a), Sequential SARFA using dataset actions (b), and Sequential SARFA using optimal actions (c). The resulting saliency maps (d) show how Sequential SARFA captures more and less piece saliency by simulating future actions. Sequential SARFA evaluated with a depth of 3, and discount factor of 0.9. Q-values were calculated based on Stockfish 15 [11] with a runtime time limit of 2.0 seconds

are vital later in the agents strategy and are highlighted in Figure 5c but not in Figure 5b.

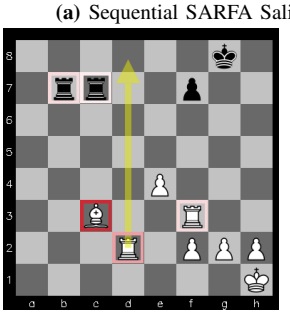
D. PaIRS

The pairwise importance method was tested on 10 unique game boards, and it attempts to identify groups of related features (i.e. chess pieces) in the current state. One of the key parameters for this method is the comparison technique used to compute the pairwise sensitivity between two pieces. The two techniques include one that calculates the absolute difference between the q-value distributions associated with the individual perturbations and another that finds the KL-divergence between their respective action distributions. Experiments with both techniques show quite different results, each with their own failure and success modes.

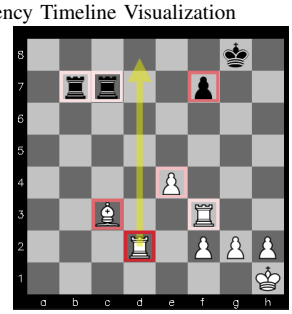
Figure 6 shows two different boards analyzed using both sensitivity techniques. In the board from 6a-b, it is white to move and the top move is to slide the rook from d3 to d8 to check the black king. The black bishop can take the rook, and the white queen can capture back with another check. The black knight would then move to e8 to block the check while defended by the black queen on c6. Essentially, the black queen, bishop, and knight all work together to prevent checkmate on the back rank.

This is one of the few cases where the KL-divergence technique is able to pick up on a grouping of pieces that serve the same functional purpose in the near future, as shown by the yellow coloring in Figure 6a. However, the technique fails to find any meaningful groupings in most situations as shown by the seemingly arbitrary red groupings in 6a and 6c. Similar

(a) Sequential SARFA Saliency Timeline Visualization



(b) SARFA Saliency



(c) Sequential SARFA Result

Fig. 5: Qualitative analysis of FEN 6k1/1rr2p2/8/8/4P3/2B2R2/3R1PPP/7K w - - 0 1. Sequential SARFA (with a depth of 5) is able to capture the importance of F3 and E4. E4 is used to stop the king in timestep 5 from entering F5 and is extremely important to the current strategy of the model.

(a) KL-divergence on actions

(b) Difference in q-values

(c) KL-divergence on actions

(d) Difference in q-values

Fig. 6: PaIRS grouping results on two different boards. (a) and (c) show the inconsistent results for the KL-divergence technique while (b) and (d) show the ability of the q-value difference technique to capture global piece values

findings are seen across several boards shown in the appendix.

On the other hand, a pattern emerges when using the technique involving measuring the difference in q-values. In chess, it is commonly known that pawns are worth 1 point, bishop and knights are worth 3 points, rooks are worth 5 points, and the queen is worth 9 points. In Figure 6b and 6d, we can see the method does a good job of separating the white and black pieces while also clustering together similarly valued pieces. This is also consistent with the results from the other boards in the appendix when using this technique.

While it may seem obvious that similar piece types should be in the same groups, chess is a unique case where these groupings are easily picked out. If extended to other environments, this method can help pick up on feature dependencies that tend to be more ambiguous. For example, this technique could highlight which financial metrics are given the same value or weight for a stock-trading agent. Therefore, using PaIRS with q-value differentials could be a great step towards further explainability and understanding which features RL-agents weight similarly when making decisions. Note that unlike the first technique, this approach focuses more on the common global value of features in a certain state. By running KL-divergence on the action distribution in the prior technique, however, we attempt to explain which features are related due to their connection to a specific behavior or action.

V. Discussion and Analysis

Saliency maps are commonly used to highlight the most important features regarding specific outputs of a neural network, and the SARFA paper [4] applies this concept to help explain why agents make certain moves in games like chess. The proposed methods in this paper show some success in

improving explainability for chess by addressing the shortcomings of SARFA.

As seen in section 4A, additive perturbations accurately identifies empty spaces on the chess board that are crucial to explaining the current move. The results are evidence that empty spaces are valuable in explaining move choice when combined with removal perturbations. There is limited literature discussing "empty saliency" but it is highly relevant for many adversarial multiplayer games. Despite the strong results, additive perturbations highlight excessive positions leading to imprecise explainability. SARFA struggles with this and is a source of overall improvement in gradient-free RL explainability. As discussed earlier, the current method for additive perturbation only uses pawns. Future work would attempt to expand the universe of pieces considered, a generalized solution that can be deployed to more environments.

SARFA struggles with explaining why certain actions were not taken. We addressed this through defensive saliency which show stro ng results in distinguishing which individual pieces are impacted by agent tactics. Separation of piece roles into offensive and defense provide an additional level of understanding past saliency. Offensive and defensive categories are easily transferrable to other games and environments.

Sequential SARFA shows promise in being able to communicate long-term strategies better than SARFA. As shown in Figure 5, Sequential SARFA provides a saliency map that explains the next action in the context of the overall strategy. As explained before, the quantitative analysis shows lackluster performance due to the dataset not being optimized for this goal. Our qualitative analysis clearly identifies situations where Sequential SARFA saliency maps better provide more context regarding the strategy beyond the next action. In future work we hope to build a stronger dataset for Sequential SARFA analysis.

Finally, the PaIRS method takes a unique approach to increasing explainability with saliency maps by attempting to identify groups of related features. This would directly address an existing issue with saliency maps which often highlight many seemingly unrelated features with no explanation for how they are dependent on each other. As discussed in section 4D, the variation of PaIRS that computes sensitivity using a KL-divergence on action distributions aims to group features that serve a similar functional purpose but fails to do so consistently. This is likely due to high variance and complexity in action probability distributions especially for games like chess. However, the results from the q-value difference technique seem to be encouraging as it is successfully able to group features with similar global value, aligning with human intuition about piece values. This success can likely be attributed to the fact that q-value distributions are fundamentally different than action distributions in the way that they capture some baseline global value for how good a state is. That also means it strays away for providing action-specific explanations which was the main weakness of this technique. Regardless, it would be interesting to see whether this ability to capture global feature dependencies would extend to other RL environments and to find ways to combine this global focus with local, action-specific explanations.

Overall, these approaches take a big step towards improving and opening up possibilities for saliency maps such that they can be at the forefront of explainability in AI. The main limitation of current methods is that even if they are able to capture all salient features, they leave humans with a lot to decipher regarding why each feature is salient. Additive perturbations and sequential SARFA try to do a better job at capturing all types of saliency while our offense/defense and PaIRS approaches break down the roles of each feature. Looking at ways to further categorize these roles based on more relevant descriptors is a promising direction. Nonetheless, using only feature saliency maps has inherent limitations and other directions could yield more fruitful results. One possibility for future research could include analyzing goal saliency for each action. Tying together goals generated by human intuition with analysis of behaviors of deep RL agents could be a major leap in AI explainability.

In the future, we would also like to evaluate an empty space perturber which identifies significant empty spaces the agent is considering. Unfortunately, Chess Saliency SARFA [4] does not include labels for empty spaces. We believe an LLM can be leveraged to generate a diverse dataset set of boards and the associated labels for the important empty spaces. This is outside the scope of our project, so we leave this as a future area of exploration.

VI. Conclusion

In this paper, we have presented multiple extensions to the SARFA approach of explainable RL in order to better capture an agent's reasoning with access to limited information. Using examples from chess, we illustrate how the salience of empty features and the type of salience of a feature can be determined through a perturbation-based approach. In addition, we show how salience can be captured for multi-step strategies using Sequential SARFA and how related pieces can be grouped using the PaIRS framework. Our results clearly demonstrate that we can gain deeper insight into an agent's behavior using our approaches. All of these techniques are generalizable to various RL environments and strategic games. Code can be found here.

References

[1] N. Puri, S. Verma, P. Gupta, D. Kayastha, S. Deshmukh, B. Krishnamurthy, and S. Singh, "Llm explainable survey," *ICLR*, 2020.

[2] R. Kashefi, L. Barekatain, M. Sabokrou, and F. Aghaeipoor, "Explainability of vision transformers: A comprehensive review and new perspectives," *arXiv:2311.06786v1 [cs.CV]*, 2023.

[3] R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," *ICCV*, 2017.

[4] N. Puri, S. Verma, P. Gupta, D. Kayastha, S. Deshmukh, B. Krishnamurthy, and S. Singh, "Explain your move: Understanding agent actions using specific and relevant feature attribution," *ICLR*, 2020.

[5] J. Li, W. Monroe, and D. Jurafsky, "Understanding neural networks through representation erasure," *arXiv:1612.08220 [cs.CL]*, 2017.

[6] J. Li, X. Chen, E. Hovy, and D. Jurafsky, "Visualizing and understanding neural models in nlp," *arXiv:1506.01066v2 [cs.CL]*, 2016.

[7] Z. Wu, Y. Chen, B. Kao, and Q. Liu, "Perturbed masking: Parameter-free probing for analyzing and interpreting bert," *ACL*, 2020.
[8] H. Mohebbi, A. Modarressi, and M. Pilehvar, "Exploring the role of bert token representations to explain sentence probing results," *EMNLP*, 2021.
[9] J. Vig, "Visualizing attention in transformer-based language representation models," *ICLR*, 2019.
[10] T. Vincent, B. Belousov, C. D'Eramo, and J. Peters, "Iterated deep q-network: Efficient learning of bellman iterations for deep reinforcement learning," 2023. [Online]. Available: https://openreview.net/forum?id=9buR1UFCDh
[11] T. Romstad, M. Sostalba, J. Kiiski, and G. Linscott, "Stockfish 15," https://stockfishchess.org/, 2022.

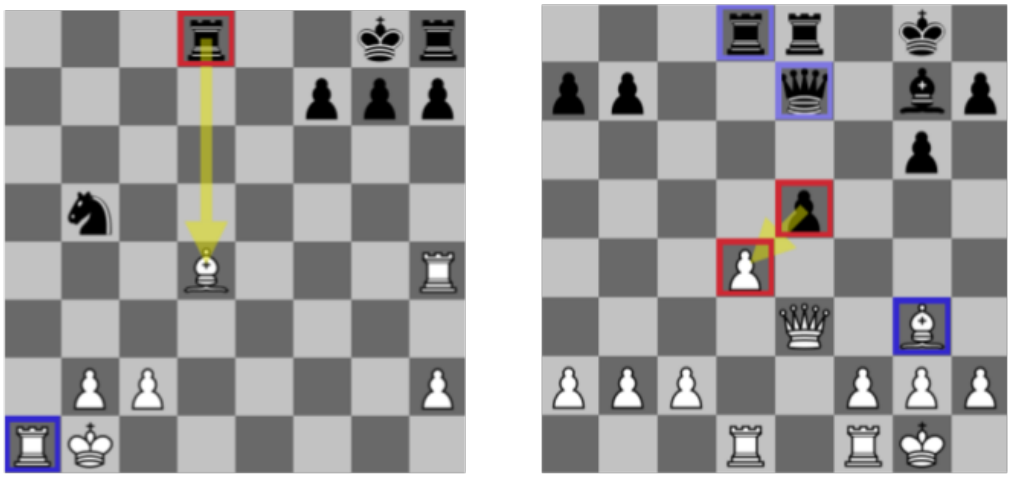


Fig. 7: Additional results of offensive and defensive saliency method. Pieces that benefit the offense are highlighted in red while pieces that benefit the defense are highlighted in blue. (a) shows a short horizon result where the white rook can deliver backrank checkmate if the black rook takes the white bishop. (b) shows a longer horizon result where the white bishop is effectively playing defense. If black takes the white pawn in the center, a queen trade would leave the black rooks open to a pin by the white bishop.

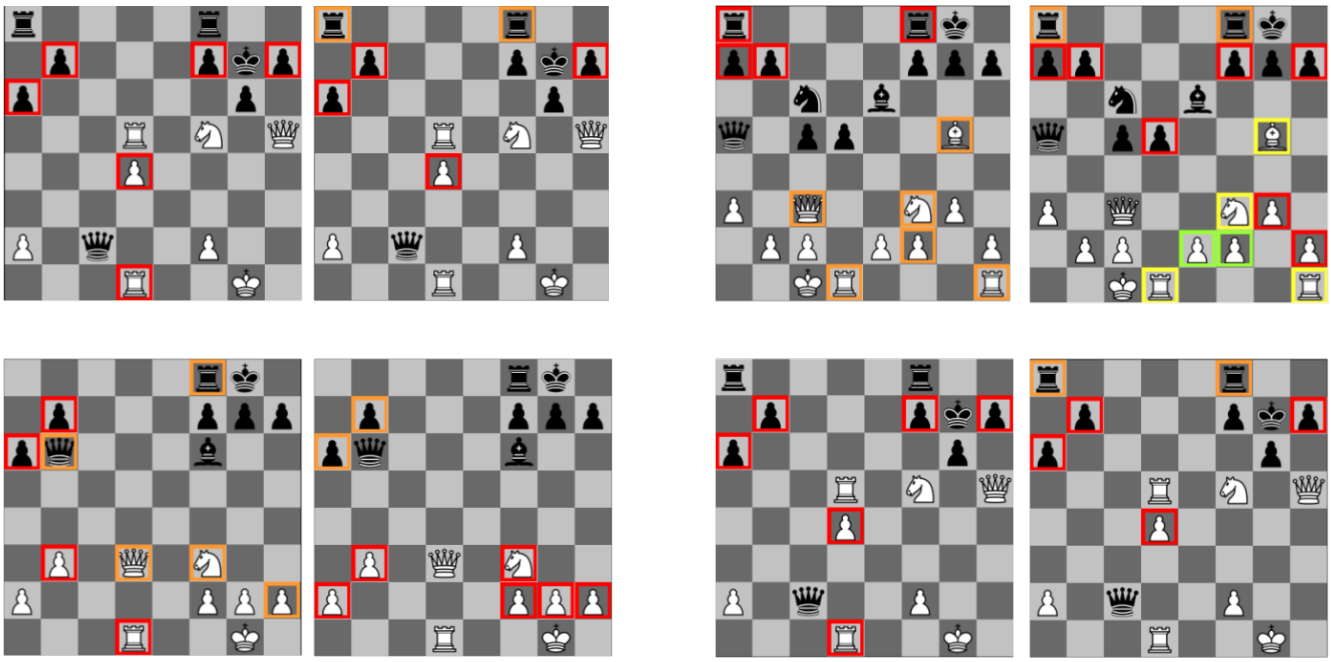


Fig. 8: Additional results for PaIRS on 4 different game boards. For each board, the left board is the groupings results from the KL-divergence on actions technique while the right board is the results from the q-value difference technique. All boards follow the same general pattern as discussed in paper earlier. The boards using the first technique show inconsistent results and noisy groupings while the second technique groups pieces based on similar global value.