Machine versus Human Attention in Deep Reinforcement Learning Tasks: Appendix

Sihang Guo
UT Austin
sguo19@utexas.edu

Ruohan Zhang
Stanford University
zharu@stanford.edu

Bo Liu
UT Austin
bliu@cs.utexas.edu

Yifeng Zhu
UT Austin
yifeng.zhu@utexas.edu,

Dana Ballard
UT Austin
danab@utexas.edu,

Mary Hayhoe
UT Austin
hayhoe@utexas.edu,

Peter Stone
UT Austin, Sony AI
pstone@cs.utexas.edu,

Appendix 1: Implementations Details

Human Gaze Prediction Models

In order to get human saliency maps for the data generated by reinforcement learning (RL) agents, we need a model that can predict human attention. We have found that the accuracy of such a model is critical for making attention comparison meaningful. Here we discuss details for implementing human attention models. The data we use is from the Atari-HEAD dataset [12][1]. The trained models and network architecture will be made available online. The prediction results for each individual game is shown in Table 1.

- Input image preprocessing: The images are reshaped from 160×210 to 84×84 with bilinear interpolation and converted into grayscale. Then we scale the pixel values to be in the range of $[0, 1]$ by dividing them by 255. So the inputs are consistent with the inputs to the reinforcement learning agents.
- Human gaze label preprocessing: Following the convention, we convert discrete gaze positions into continuous distribution by blurring each gaze location using a 2D Gaussian with σ that is equivalent to one visual degree [8, 12].
- Model architecture: The human gaze prediction model is adapted from [12]. The network has three convolution layers followed by three deconvolution layers. Their parameters are as follows:
  - Convolution layer 1: 32 filters, kernel size $= 8 \times 8$, stride $= 4$, followed by relu activation, batch normalization, and dropout.
  - Convolution layer 2: 64 filters, kernel size $= 4 \times 4$, stride $= 2$, followed by relu activation, batch normalization, and dropout.
  - Convolution layer 3: 64 filters, kernel size $= 3 \times 3$, stride $= 1$, followed by relu activation, batch normalization, and dropout.
  - Deconvolution layer 1: 64 filters, kernel size $= 3 \times 3$, stride $= 1$, followed by relu activation, batch normalization, and dropout.

[1] Available for download at: https://zenodo.org/record/3451402

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

	Breakout	Freeway	Frostbite	Ms.Pac-Man	Montezuma	Seaquest
AUC	0.973	0.977	0.962	0.984	0.947	0.964
CC	0.575	0.627	0.515	0.666	0.447	0.546
KL	1.302	1.205	1.555	1.027	1.882	1.495

Table 1: Human gaze prediction accuracy for 6 Atari games. Random prediction baseline: AUC = 0.500, KL = 6.100, CC = 0.000. The prediction accuracy is comparable to previous results [12] and is considered high according to the visual saliency research standards [1]. A gaze prediction video for all 6 games has been included in the multimedia appendix.

  - Deconvolution layer 2: 64 filters, kernel size $= 4 \times 4$, stride $= 2$, followed by relu activation, batch normalization, and dropout.
  - Deconvolution layer 3: 1 filter, kernel size $= 8 \times 8$, stride $= 4$, followed by a softmax layer.

  The network is implemented using Tensorflow 1.8.0 and Keras 2.1.5. The same deep network architecture and hyperparameters are used for all games.
- Optimizer: The optimizer is Adadelta which is a method with adaptive learning rate [10]. We use learning rate $= 1.0$, decay rate $\rho = 0.95$, and $\epsilon = 1e - 8$, batch size $= 50$, number of training epochs $= 70$.
- Data: For each game, we use approximately 80% gaze data (16 trials) for training and 20% (4 trials) for testing. For this dataset, two adjacent images or gaze positions are highly correlated. We avoid putting one frame in the training set and its neighboring frame in the testing set by using complete trials as the testing set. This makes sure that the data belonging to the same trajectory will not end up in both training and testing.
- Hardware: Training was conducted on server clusters with NVIDIA GTX 1080 and 1080Ti GPUs.

Motion and Saliency Baselines

We include two attention models in addition to human gaze. The first one captures the motion information, measured by Farneback optical flow between two consecutive images [2]. The function is implemented as follows:

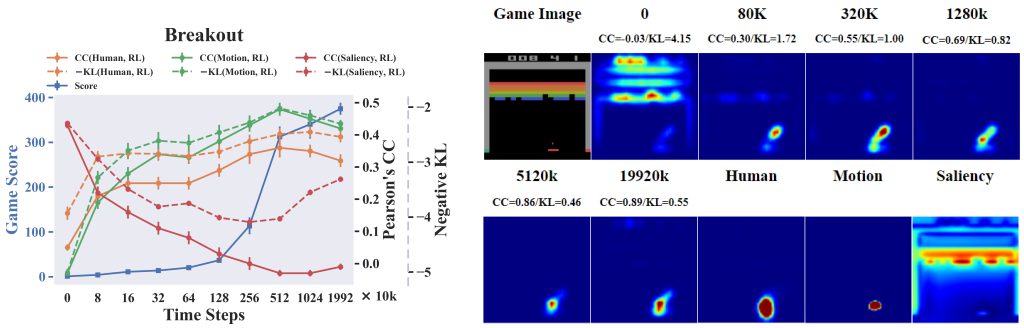
```
import cv2
import numpy as np
# The function is: calcOpticalFlowFarneback(prev_img, next_img, flow,
# pyr_scale, levels, winsize, iterations, poly_n, poly_sigma, flags)
flow = cv2.calcOpticalFlowFarneback(prev, cur, None, 0.5, 3, 15, 3, 5, 1.1,
cv2.OPTFLOW_FARNEBACK_GAUSSIAN)
# we only use the information of magnitude
fx, fy = flow[:,:,0], flow[:,:,1]
flow = np.sqrt(fx*fx+fy*fy)
```

The second model captures salient low-level image features, including color, orientation, and intensity (weighted equally), computed by the classic Itti-Koch saliency model [7]. We use the Python implementation provided by `https://github.com/akisatok/pySaliencyMap` without modification except image dimensions.

Appendix 2: The Effects of Learning on Attention

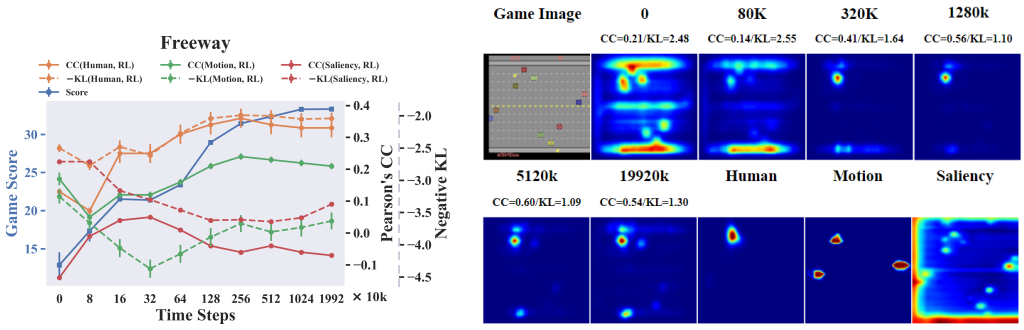
In Appendix 2 to 5, we will show statistics and example images of each game. Atari games have very different reward mechanisms, visual features, and dynamics. Hence, it is often difficult to find an RL algorithm that works best for all games. We hope that, by showing results for individual games, researchers will gain insights into why a particular algorithm (like PPO here) performs well or poorly for a particular game.

In Figure 1 to 6 we show how the attention of the RL agent (PPO) evolves over time compared to human attention. Part (a) of each figure shows the similarity metrics: Pearson's Correlation Coefficient (CC) and negative Kullback-Leibler Divergence (KL) values over training time steps. The values are averaged over 100 images in the standard image set (as described in section 3.3). (a) also shows the game scores (averaged over 50 episodes) over training time steps. Part (b) of each figure shows an example game image. It also includes the average saliency maps of the RL agents during training and a human saliency map predicted by the human model for the selected game image. Note that KL values are negated for better visualization.



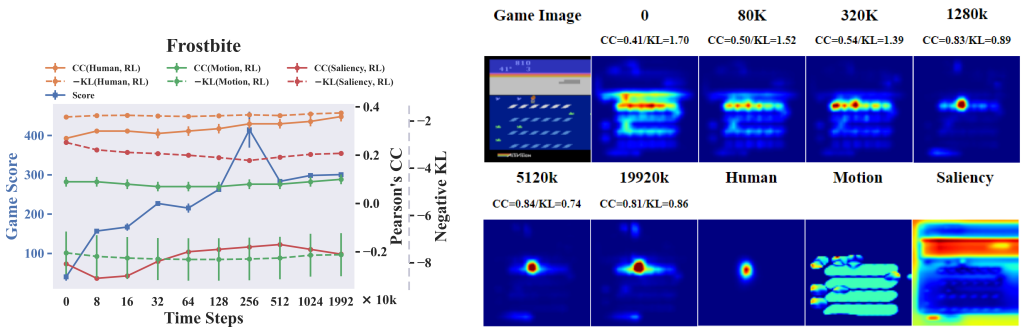
(a) (b)

Figure 1: Breakout: (a) Human and RL saliency maps become more similar over training time steps. Pearson's correlation coefficients between game score and human are CC: $r(8) = 0.664, p < 0.05$, KL: $r(8) = 0.622, p = 0.054$; between game score and motion are CC: $r(8) = 0.641, p < 0.05$, KL: $r(8) = 0.550, p = 0.100$; between game score and saliency are CC: $r(8) = -0.661, p < 0.05$, KL: $r(8) = -0.215, p = 0.551$. (b) The RL agents gradually learn to focus their attention on both the paddle and the ball as humans do.



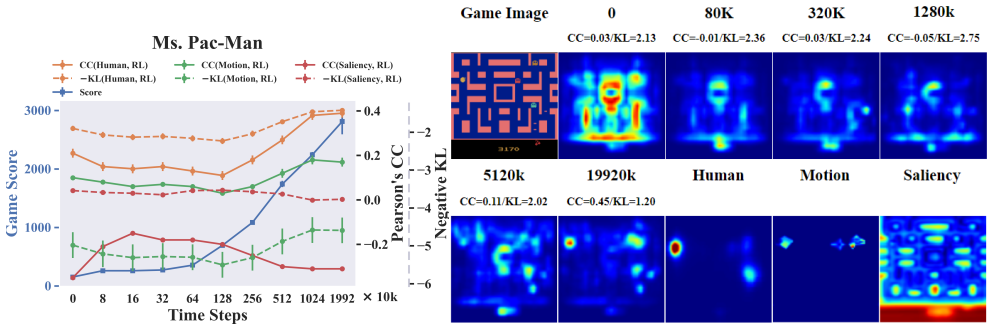
(a) (b)

Figure 2: Freeway: (a) Human and RL saliency maps become more similar over training time steps. Pearson's correlation coefficients between game score and human are CC: $r(8) = 0.878, p < 0.001$, KL: $r(8) = 0.888, p < 0.001$; between game score and motion are CC: $r(8) = 0.765, p < 0.01$, KL: $r(8) = 0.080, p = 0.826$; between game score and saliency are CC: $r(8) = -0.078, p = 0.830$, KL: $r(8) = -0.878, p < 0.001$. (b) The RL agents gradually learn to focus their attention on the yellow chicken being controlled to cross the highway. The similarity values decrease a little at the end of the training because the RL agents also learn to attend to the starting point at the bottom of the image.



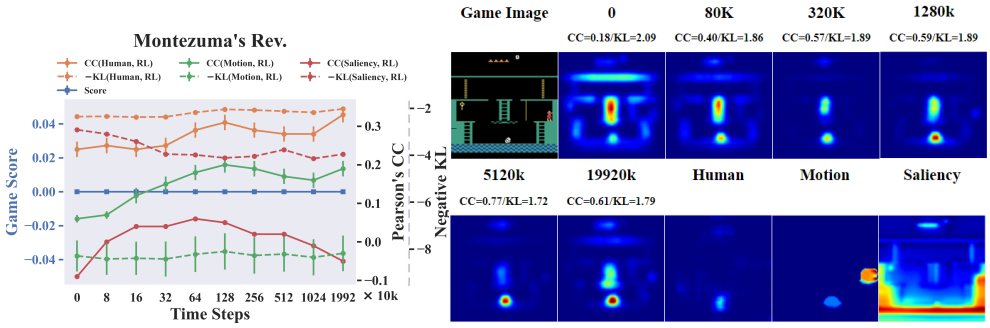
(a) (b)

Figure 3: Frostbite: (a) Human and RL saliency maps become more similar over training time steps. Pearson's correlation coefficients between game score and human are CC: $r(8) = 0.791, p < 0.01$, KL: $r(8) = 0.620, p = 0.056$; between game score and motion are CC: $r(8) = -0.087, p = 0.811$, KL: $r(8) = -0.443, p = 0.200$; between game score and saliency are CC: $r(8) = 0.688, p < 0.05$, KL: $r(8) = -0.900, p < 0.001$. (b) The RL agents gradually learn to attend to the little person being controlled in the middle like humans do.



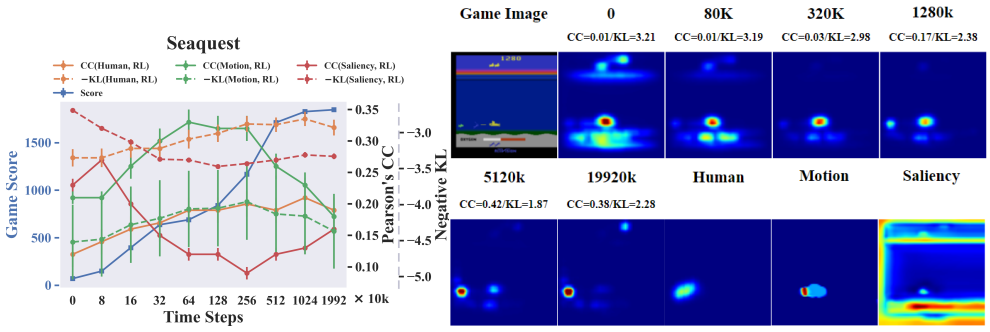
(a) (b)

Figure 4: Ms. Pac-Man: (a) Human and RL saliency maps become less similar at first, and then become more similar during training. Pearson's correlation coefficients between game score and human are CC: $r(8) = 0.910, p < 0.001$, KL: $r(8) = 0.893, p < 0.001$; between game score and motion are CC: $r(8) = 0.819, p < 0.01$, KL: $r(8) = 0.809, p < 0.01$; between game score and saliency are CC: $r(8) = -0.586, p = 0.075$, KL: $r(8) = -0.806, p < 0.01$. (b) The RL agents eventually learn to attend to the Pac-Man on the left and an enemy ghost on the right like humans do.



(a) (b)

Figure 5: Montezuma's Revenge: (a) Human and RL saliency maps becomes more similar over training time steps. Note that this is a difficult game for RL agents and they never learn to score. Pearson's correlation coefficients are undefined in this case. (b) The RL agents learn to attend to the enemy at the bottom like humans do, but they are uncertain about the importance of the ladder in the middle.

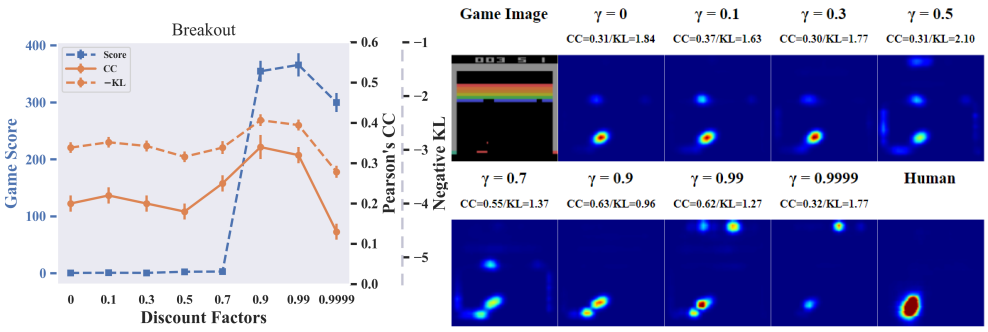


(a) (b)

Figure 6: Seaquest: (a) Human and RL saliency maps becomes more similar according to the KL metric (yellow dashed line). Pearson's correlation coefficients between game score and human are CC: $r(8) = 0.824, p < 0.01$; KL: $r(8) = 0.930, p < 0.001$; between game score and motion are CC: $r(8) = -0.116, p = 0.750$, KL: $r(8) = 0.419, p = 0.228$; between game score and saliency are CC: $r(8) = -0.653, p < 0.05$, KL: $r(8) = -0.641, p < 0.05$. (b) The RL agents learn to attend to an incoming enemy on the left.

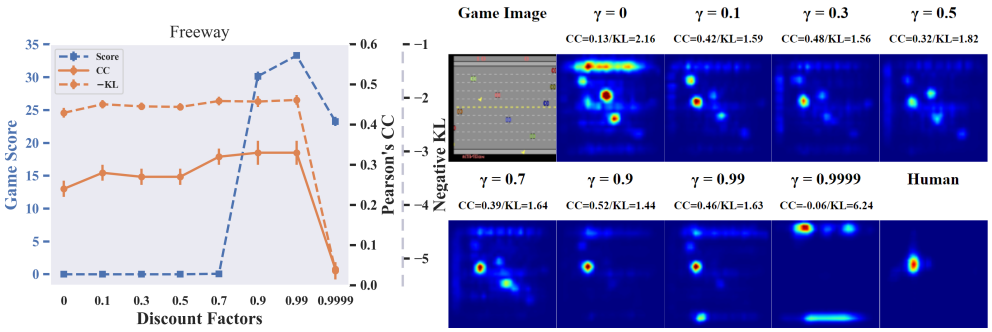
Appendix 3: The Effects of Discount Factors on Attention

Similar to Appendix 2, Figure 7 to 12 shows how the attention of the RL agent (PPO) changes when we vary the discount factor, compared to human attention. $\gamma = 0.99$ is the default value for most RL algorithms [5, 9]. Each Figure (a) shows the similarity metrics: CC and negative KL values over different discount factors. The values are averaged over 100 images in the standard image set. (a) also shows the game scores (averaged over 50 episodes) over discount factors. Each Figure (b) shows an example game image, RL agents' saliency maps with different discount factors γ, and human saliency map predicted by the human model. Note that KL values are negated for better visualization.



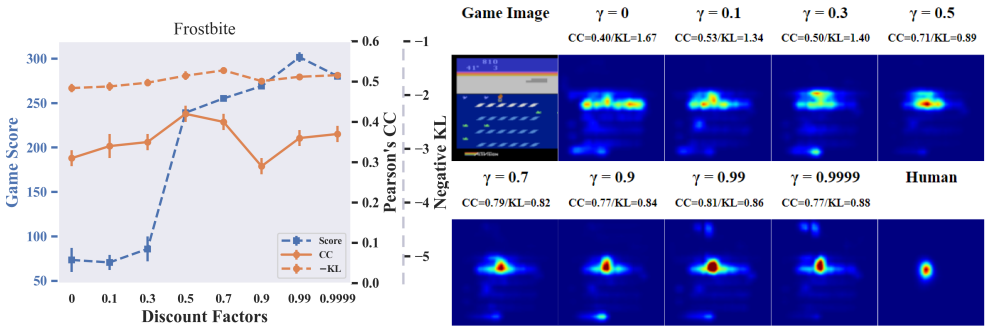
(a) (b)

Figure 7: Breakout: (a) The RL agent's attention is most similar to human's when $\gamma = 0.9$. (b) Human attention is on the paddle and the ball. Setting $\gamma > 0.9$ makes the agent attend to the score at the top of the image.



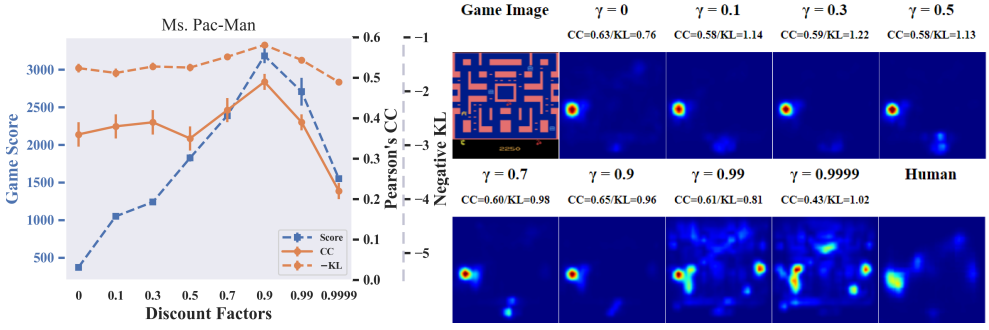
(a) (b)

Figure 8: Freeway: (a) The RL agent's attention is most similar to human's when $\gamma = 0.9$ and $\gamma = 0.99$. (b) Human attention is on the yellow chicken being controlled to cross the highway.



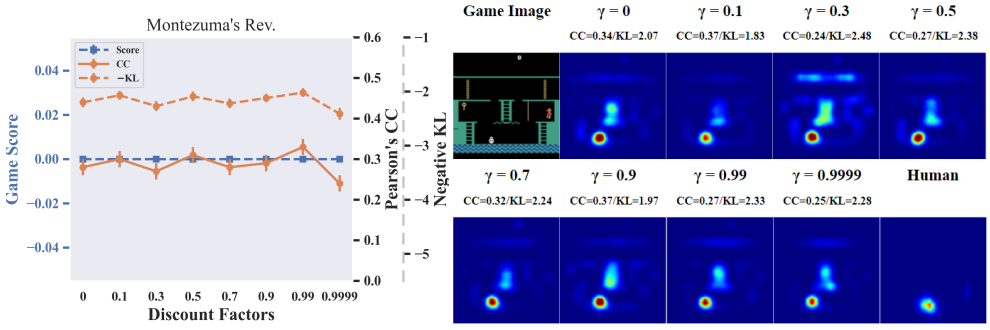
(a) (b)

Figure 9: Frostbite: (a) The RL agent's attention is most similar to human's when $\gamma = 0.7$ (CC) or 0.5 (KL). (b) Human attention is on the little person being controlled in the middle.



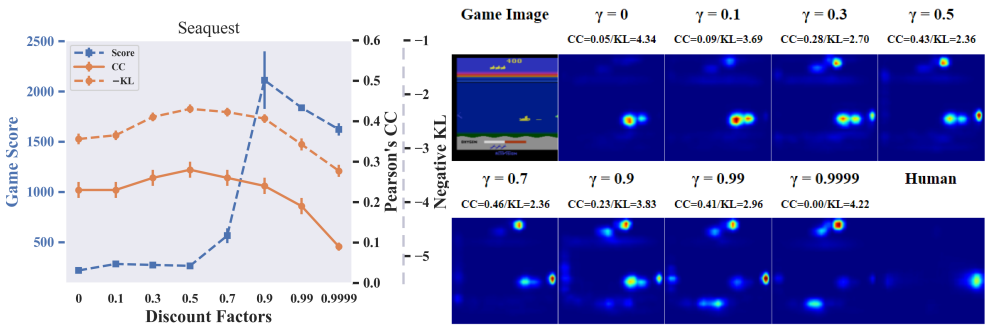
(a) (b)

Figure 10: Ms. Pac-Man: (a) The RL agent's attention is most similar to human's when $\gamma = 0.9$. Note that choosing this value and deviating from the default $\gamma = 0.99$ lead to a better performance. (b) Human attention is mostly on the Pac-Man on the left side. Setting $\gamma > 0.9$ distracts the agent to attend to other objects.



(a) (b)

Figure 11: Montezuma's Revenge: (a) The RL agent's attention is most similar to human's when $\gamma = 0.99$. Note that this is a difficult game for RL agents and they never learn to score. (b) Human attention is on the enemy at the bottom.



(a) (b)

Figure 12: Seaquest: (a) The RL agent's attention is most similar to human's when $\gamma = 0.5$. Note that choosing $\gamma = 0.9$ and deviating from the default $\gamma = 0.99$ lead to a better performance. (b) Human attention is on an appearing enemy on the right side. With $\gamma > 0.9$ the RL agent also learns to attend to the oxygen bar at the bottom.

Appendix 4: Failure States Analysis

Figures 13 and 14 show RL agents' saliency maps compared to human's in failure states. These states are game frames right before the RL agent loses a "life" which incurs a large penalty in Atari games. This analysis is helpful in answering the question: Did RL agents make mistakes because they fail to attend to the right objects, or did they attend to the right objects but make wrong decisions? Figure 13 shows the games that belong to the former case, and Figure 14 shows the games that belong to the latter case. Freeway is excluded here since the PPO agent learned a policy that is nearly optimal.

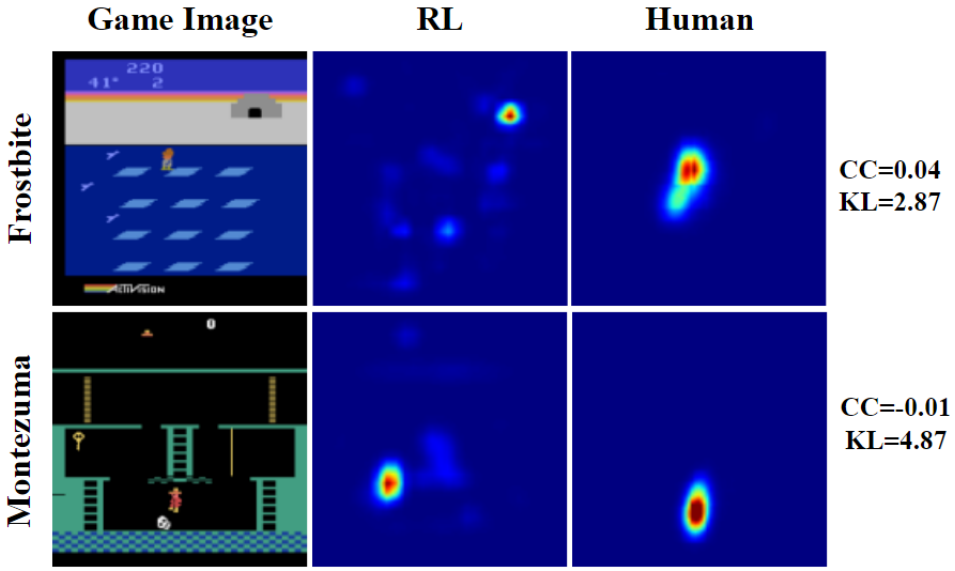


Figure 13: Games in which human attention and RL agents' attention are more different in the failure states than the normal states. This indicates that in these games the mistakes are likely caused by wrong attention which subsequently led to wrong decisions. Frostbite: The RL agent is attending to the entrance of the Igloo. It should attend to the little person in the middle like humans do to avoid an incoming enemy from the left. Montezuma's Revenge: The RL agent is attending to the bottom of the ladder. It should attend the little person and the enemy to escape from the dangerous situation.

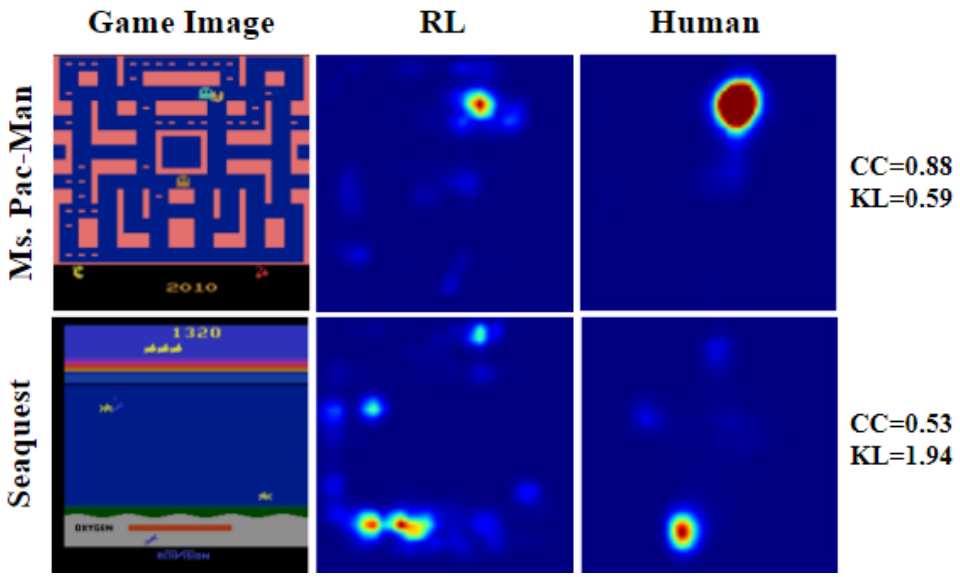


Figure 14: Games in which human attention and RL agents' attention are more similar in the failure states than the normal states. This suggests that they generally agree on the objects to be attended to. But the RL agents made wrong decisions due to its suboptimal policy. Ms.Pac-Man: The agent and the human both attend to the Pac-Man which is about to be captured by the cyan enemy ghost. The agent failed to run away from it. Seaquest: The agent and the human both attend to the empty oxygen bar at the bottom. The agent failed to refill oxygen before it runs out.

Appendix 5: Generalizing to Unseen Data

This sections lists the parameters and figures when comparing RL agents' and human's saliency maps in unseen states. The unseen states are late-game states obtained from human experts' which RL agents have not encountered (above the agents' best score). The goal is to see whether RL agents' attention can reasonably generalize to these unseen states. Table 2 shows the score threshold for choosing late-game states in each game and Figure 15 to 16 show RL agents' saliency maps compared to human's in the unseen states. Again, Freeway is excluded here since the PPO agent learned a policy that is nearly optimal.

Game	Score Threshold
Breakout	350
Freeway	-
Frostbite	320
MontezumaRevenge	0
MsPacman	3000
Seaquest	1800
SpaceInvaders	2000

Table 2: Game score thresholds for choosing the unseen dataset. Frames in the unseen dataset are randomly chosen from the subset of all frames with scores above the threshold. Note that Freeway is excluded in the analyses because the PPO agent learned a policy that is nearly optimal; Montezuma's Revenge has a threshold score 0 becasue the PPO agent never learned to score in this game.

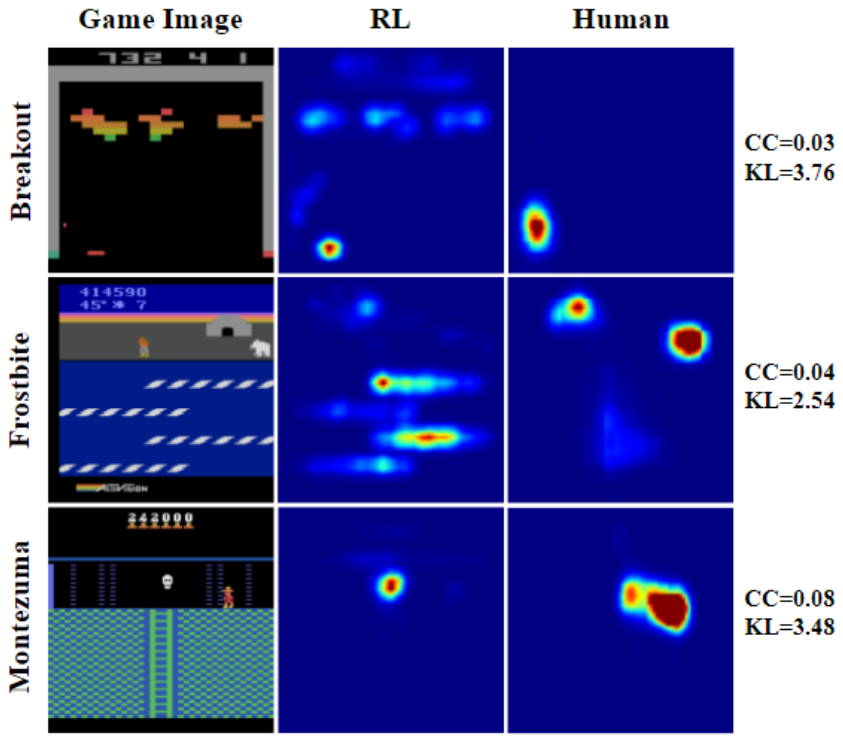


Figure 15: Games in which human attention and RL agents' attention are more different in unseen states than the normal states. This is mostly due to new objects that the agents have never encountered. Breakout: The CC value drops significantly due to unseen spatial layouts of the bricks. The KL does not change much because there are no new objects so the agent can still attend to human attended objects like the ball on the left. Frostbite: Human attention is around the polar bear (a new object) at the upper right corner. Montezuma's Revenge: Human attention is on the fire beacon (a new object). The RL agent's attention is on the skull which is a familiar object.

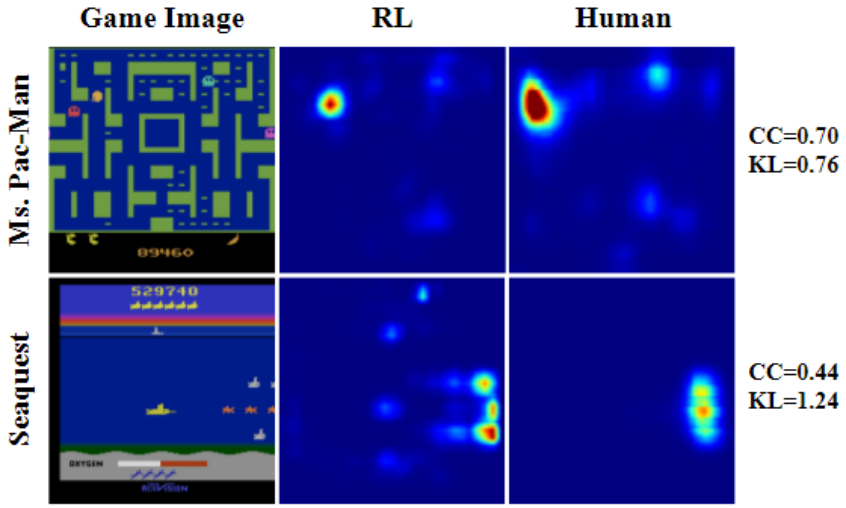


Figure 16: Games in which human attention and RL agents' attention are more similar in the unseen states than the normal states. This is because there are no new objects in these unseen states – objects move much faster and appear in larger numbers. The player often encounters dangerous states that are close to failure. As shown in Appendix Fig. 14 human attention and RL agent's attention are often similar in failure states for these two games. Ms. Pac-Man: The agent and the human both attend to the Pac-Man which is about to be captured by the red enemy ghost. Seaquest: The agent and the human both attend to the enemies on the right side.

Appendix 6: Other Deep RL Algorithms

There is a positive correlation between model performance (in terms of the game score, averaged over 50 episodes each) and similarity (in terms of CC on the standard image set) with human attention. Fig. 17 shows the result for Breakout, Freeway, Frostbite, and Seaquest (Ms.Pac-Man's result is in the main text). Montezuma's Revenge is excluded because most algorithms have a score of zero. Note algorithms such as DQN outputs state-action values instead of policy hence they will attend to game scores at the top or the bottom of the screen. In order to ensure a fair comparison for policy-based and value-based algorithms, we ignore attention that is on game scores when doing comparisons (by cropping out that part of the image when calculating the similarity), following the standard approach [3, 4, 6].

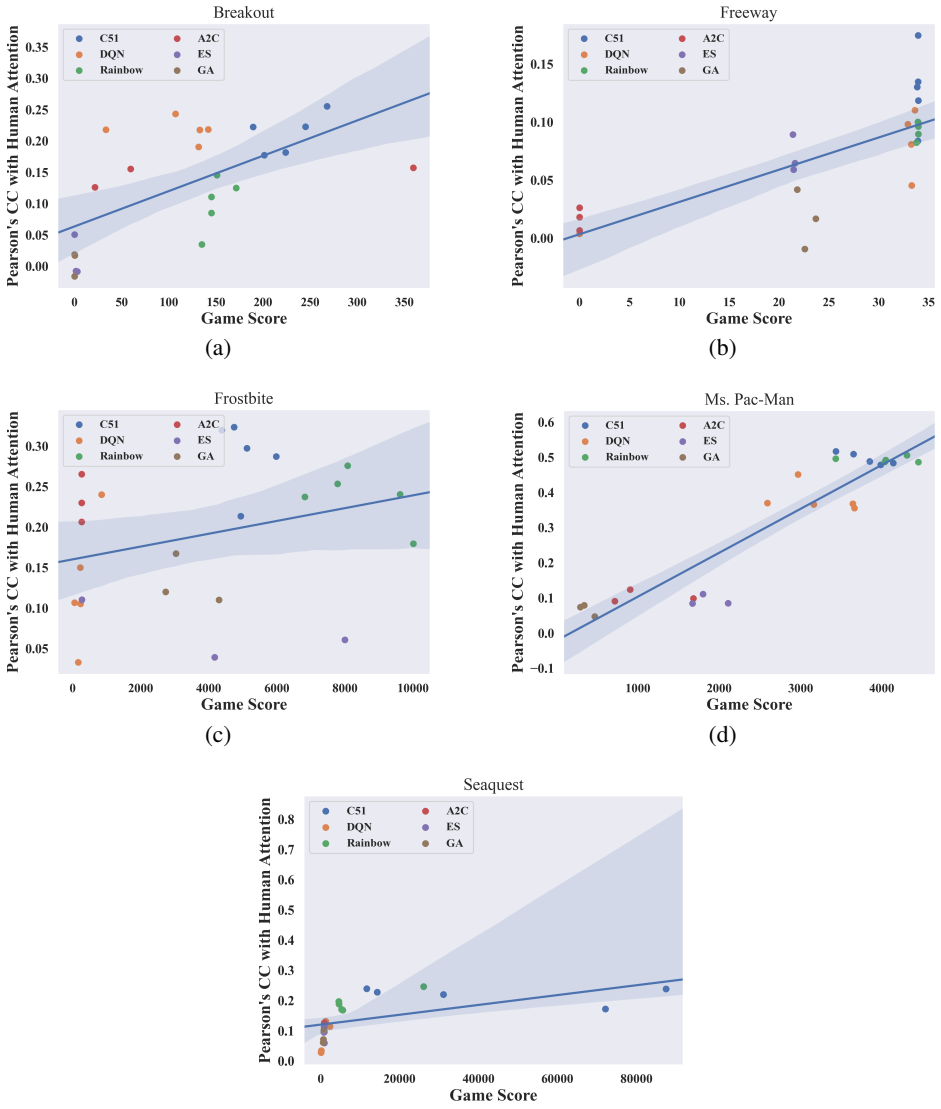


Figure 17: The relation between the similarity with human attention and algorithm's performance. The line shows the linear regression line fitted to the data point and the shaded area is the 95% confidence interval. The correlation coefficients between the similarity measurement and game score are: $r(22) = 0.634, p < 0.001$ for Breakout, $r(22) = 0.743, p < 0.001$ for Freeway, $r(22) = 0.298, p = 0.158$ for Frostbite, $r(22) = 0.927, p < 0.001$ for Ms. Pac-Man, and $r(22) = 0.553, p < 0.01$ for Seaquest.

Appendix 7: The Effects of Learning and Discount Factors on Attention using Human Data

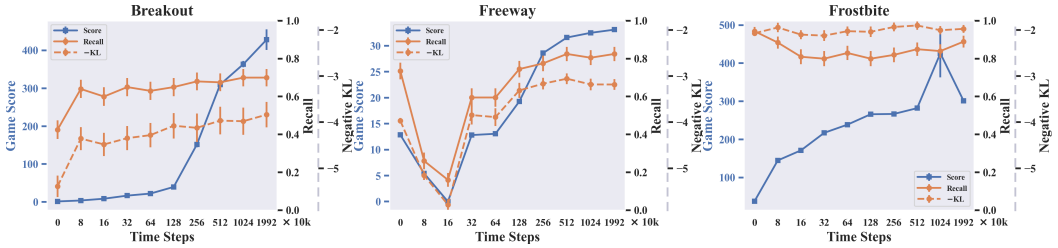
In this work, for experiments that we can use either RL or human data, we focused on the explainable RL issue and performed the analyses using RL data in the main text. Here we have performed the same analyses on the effects of learning and discount factors using human states and eye fixation data to confirm that using human data does not affect our main conclusions.

Instead of using raw gaze positions as in all the previous works [11, 12], we extract human fixation locations using the software provided by the Eyelink eye tracker[2]. Fixational eye movements are to maintain current gaze location by fixating our eyes onto and inspect the object of interest. Therefore fixations are more meaningful for indicating human visual attention. Since fixations are discrete gaze locations, *recall* could be used to compare pixel-base human attention maps. It is defined by the following formula:

$$\text{Recall}(P, Q) = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \tag{1}$$

where a pixel is counted as a positive prediction if the normalized RL attention value is greater than the probability of randomly choosing a pixel from the frame (in our case, $1/(84 \times 84)$, where all attention maps are in size 84×84), and negative prediction otherwise. For the KL comparison, we treated human fixation maps as distributions by placing Gaussian blurs centered at the fixation location, with σ of one to two visual degrees of the human visual field.

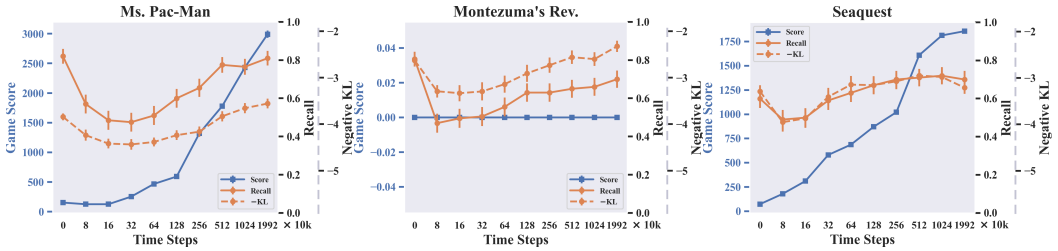
The results are shown in Figures. 18 and 19. By comparing these results to those presented in Appendix 3 and 4, we show that using human data does not affect our main conclusions. The RL agents' attention becomes more similar to humans' during the learning process. The agents' attention was most similar to human attention around $\gamma = 0.9$.



(a) Breakout. Pearson's Correlations: Recall: $r = 0.58$, $p = 0.08$; KL: $r = 0.64$, $p = 0.04$.

(b) Freeway. Pearson's Correlations: Recall: $r = 0.88$, $p < 0.001$; KL: $r = 0.90$, $p < 0.001$.

(c) Frostbite. Pearson's Correlations: Recall: $r = -0.46$, $p = 0.17$; KL: $r = 0.34$, $p = 0.32$.



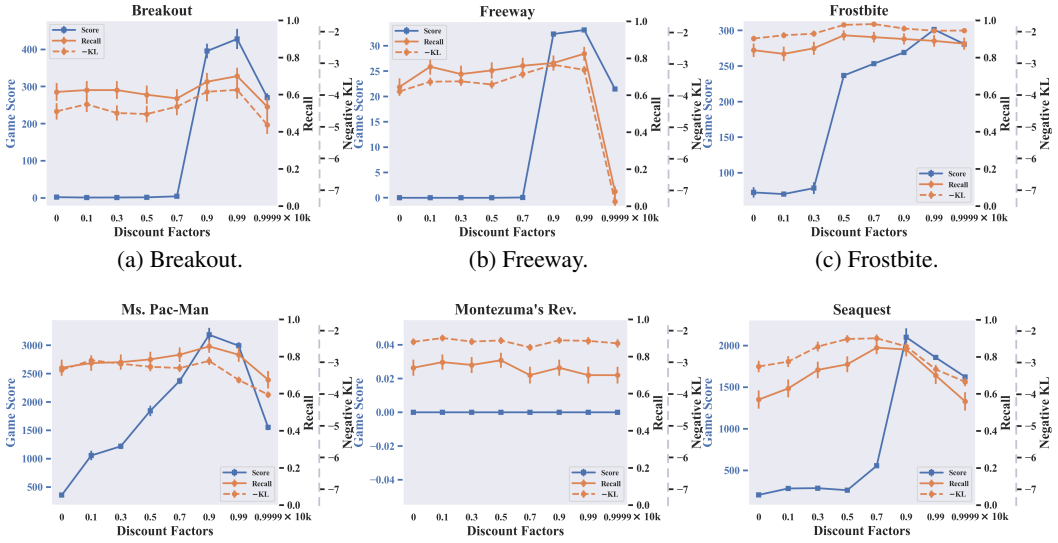
(d) Ms. Pac-Man. Pearson's Correlations: Recall: $r = 0.68$, $p = 0.03$; KL: $r = 0.82$, $p = 0.003$.

(e) Montezuma's Revenge. Pearson's correlation coefficients are undefined due to zero scores.

(f) Seaquest. Pearson's Correlations: Recall: $r = 0.84$, $p = 0.002$; KL: $r = 0.70$, $p = 0.026$.

Figure 18: The agents' game performance and attention similarity plotted against training timesteps. In general, the agents' attention becomes more similar to humans' as training progresses. The results are similar to the results presented in Appendix 2 using data generated by RL agents.

[2] Available at https://www.sr-research.com/data-viewer/



(a) Breakout. (b) Freeway. (c) Frostbite.

(d) Ms. Pac-Man. (e) Montezuma's Revenge. (f) Seaquest.

Figure 19: The agents' game performance and attention similarity plotted against discount factors. In general, the agents' attention was most similar to human attention around $\gamma = 0.9$. The results are similar to the results presented in Appendix 3 using data generated by RL agents.

References

[1] Zoya Bylinskii, Tilke Judd, Aude Oliva, Antonio Torralba, and Frédo Durand. What do different evaluation metrics tell us about saliency models? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(3):740–757, 2019.

[2] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. *Image analysis*, pages 363–370, 2003.

[3] Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. In *International Conference on Machine Learning*, pages 1792–1801, 2018.

[4] Piyush Gupta, Nikaash Puri, Sukriti Verma, Sameer Singh, Dhruv Kayastha, Shripad Deshmukh, and Balaji Krishnamurthy. Explain your move: Understanding agent actions using focused feature saliency. *arXiv preprint arXiv:1912.12191*, 2019.

[5] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018.

[6] Tobias Huber, Benedikt Limmer, and Elisabeth André. Benchmarking perturbation-based saliency maps for explaining deep reinforcement learning agents. *arXiv preprint arXiv:2101.07312*, 2021.

[7] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (11):1254–1259, 1998.

[8] Olivier Le Meur and Thierry Baccino. Methods for comparing scanpaths and saliency maps: strengths and weaknesses. *Behavior research methods*, 45(1):251–266, 2013.

[9] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[10] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[11] Ruohan Zhang, Zhuode Liu, Luxin Zhang, Jake A Whritner, Karl S Muller, Mary M Hayhoe, and Dana H Ballard. Agil: Learning attention from human for visuomotor tasks. In *European Conference on Computer Vision (ECCV)*, pages 692–707. Springer, 2018.

[12] Ruohan Zhang, Calen Walshe, Zhuode Liu, Lin Guan, Karl S Muller, Jake A Whritner, Luxin Zhang, Mary M Hayhoe, and Dana H Ballard. Atari-head: Atari human eye-tracking and demonstration dataset. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 2020.