Human-Machine Cooperation through Human-Like Visual Search Model

Aditya Acharya
a.acharya.1@bham.ac.uk
School of Computer Science
University of Birmingham

Chris Baber
c.baber@bham.ac.uk
School of Computer Science
University of Birmingham

Leonardo Stella
l.stella@bham.ac.uk
School of Computer Science
University of Birmingham

Andrew Howes
andrew.howes@exeter.ac.uk
Department of Computer Science
University of Exeter

Abstract

Human-machine teaming is a challenging and important problem that requires designing autonomous agents that can effectively cooperate with humans in complex and dynamic scenarios. This paper explores multi-agent dynamics in a reinforcement learning (RL) framework in a Stag Hunt scenario, where interactions can be either cooperative or independent. We design a system involving two agents: Agent A, which follows a mixed strategy based on predefined probability distributions, and Agent B, an RL agent with human-like visual constraint that learns an adaptive strategy to selectively sample information from the environment to infer Agent A's intent. Our results indicate that Agent B adapts its policy effectively, exhibiting adaptive gaze strategies tailored to Agent A's policy. We discuss the implications of our findings and the design of RL agents capable of interacting with human-like agents.

1 Introduction

Most Multi-Agent Reinforcement Learning (MARL) approaches do not consider the presence and influence of human teammates. This limits the applicability of RL to those real-world scenarios in which humans and machines collaborate to achieve common goals, e.g., in search and rescue missions (Moosavi et al., 2024), autonomous robots work alongside human rescuers, adapting to their intentions while communicating their own plans and actions.

Human-machine teaming is a crucial challenge for developing effective and trustworthy autonomous systems (McNeese et al., 2018). Designing agents that can interact with humans naturally and flexibly requires the agents to use models that reflect human decision-making. Our approach to creating such models is based on the assumption that people search for information that is relevant to their decisions. This assumption parallels the work of (Gigerenzer, 2002) and requires information on the policy governing information search and use. From this, our models use visual search parameters that are drawn from human performance and defined in terms of resource rationality (Howes et al., 2009; Lieder & Griffiths, 2020), i.e., constrained by the assumption that the information search is performed to seek sufficient (not complete) information from an environment.

In a static environment, this task involves determining the location of targets that relate to a specific goal (Xiuli et al. (2017); Howes et al. (2018); Cheema et al. (2020); Chen et al. (2021); Jokinen et al. (2021)). When the environment is dynamic or contains other agents, the model needs to track the movement of potential targets and infer the possible intent of the other agents. Such intent could relate to their goal state, e.g., based on their history of movements or physical proximity to targets. In this paper, we demonstrate that extending the resource rational approach to information search makes it possible to create a human-like model that responds to dynamic environments containing other agents.

To evaluate our model, we draw inspiration from game theory, which provides formal models and empirical studies of how humans and animals behave in situations where the outcome depends not only on their own actions but also on the actions of others. In particular, the Stag Hunt game, originally proposed by Rousseau (1755), is a classic example of a coordination dilemma, where agents must choose between cooperating for a higher collective payoff or acting independently for a lower but safer individual payoff. Unlike the prisoner's dilemma, where defection is always the dominant strategy, Stag Hunt has two Nash equilibria: both agents cooperate or both agents defect. However, the former equilibrium is Pareto-efficient, meaning that no agent can improve their payoff without making the other agent worse, while the latter is not. Therefore, the optimal strategy depends on the preferences and beliefs of the agents and the information available to them. In a one-shot version of Stag Hunt, the only information available to the decision-maker is the payoff matrix. The intention of the opponent is unknown. In an iterative version of the game, the opponent's intention could be inferred from the history of their previous choices. In a grid-world version, the opponent's intention could be inferred from their location and movement. This paper proposes an RL framework for studying human-machine teaming in a grid-world Stag Hunt scenario. Our main contributions and objectives are as follows:

- We introduce an RL framework for human-machine teaming in a Stag Hunt scenario to model the interaction between two agents with different characteristics and abilities.
- We design and implement an RL agent with human-like visual constraint that learns to selectively sample information from the environment to infer the intent of another agent and to cooperate or defect accordingly.

2 Related Work

Computational modelling of human eye movements has a rich history, marked by diverse theoretical frameworks. Heuristic models, such as those proposing salience maps Itti & Koch (2000) and activation maps Wolfe (2007), suggest that saccades bring the fovea, the area of sharpest vision, into alignment with regions that stand out. Approaches rooted in Bayesian theory Myers et al. (2013); Najemnik & Geisler (2008) assume that saccades target regions where visual acuity is low, effectively gathering additional information to update a Bayesian estimation of the visual environment. This framework implies that eye movements are strategic, aiming to optimise the information gained from each saccade. Optimal control models Butko & Movellan (2008); Nunez-Varela & Wyatt (2013); Hayhoe & Ballard (2014); Mnih et al. (2014); Howes et al. (2018); Chen et al. (2021) regard saccades as actions taken to maximise task-specific utility or reward. These models incorporate a cost-benefit analysis, suggesting that the programming of saccades is influenced by the task demands and the potential rewards of focusing on different areas of the visual field. This paper extends the optimal control framework to a more challenging setting where the world is dynamic and the task involves interacting with an opponent.

3 Task

We adopt the stag hunt environment from (Peysakhovich & Lerer, 2018). In this environment, two agents can choose to either cooperate or defect in each round. If both agents cooperate, they receive a high reward ($R = 5$), representing hunting a stag together. If both agents defect, they receive a low reward ($R = 1$), representing foraging a plant alone. However, if one agent cooperates and the other defects, the cooperator receives a penalty reward ($R = -5$) for hunting a stag alone, and the defector receives a low reward ($R = 1$) for ignoring the stag and foraging a plant alone. The payoff matrix for this environment is shown below in Table 1.

	cooperate	**defect**	
cooperate	(5, 5)	(-5, 1)	p = 0.8
defect	(1, -5)	(1, 1)	1-p

Table 1: Stag hunt environment payoff matrix

3.1 Implementation Details

We implement our environment in Python using the open-source code base by (Nesterov-Rappoport, 2022). The environment consists of a grid world with 5 x 5 cells (see 'Game Environment' in Figure 1), each representing a possible location for the agent or the prey, and displayed as an image of size 160 x 160 pixels or 4 x 4 degrees visual angle.

At the beginning of each game episode, the stag is randomly placed in one of the unoccupied cells. The stag remains stationary for the duration of the episode. The agents are initialised at fixed locations near the top corners of the grid. The plants are randomly distributed over the remaining cells, with a fixed density of 2.

The agents have five possible actions: MOVE UP, MOVE DOWN, MOVE LEFT, MOVE RIGHT, or STAND. Each action moves the agent to an adjacent cell or keeps it in the same cell unless the target cell is out of bounds. In that case, the action has no effect, and the agent stays in the same cell. Each action incurs a small negative reward of $R_{move} = -0.1$ ($R_{stand} = 0$) to encourage efficient behaviour.

Agent A follows a predefined mixed policy of pursuing the stag or the nearest plant with probabilities p = (0.8, 0.2), respectively. An initial model had a 50:50 distribution, but this resulted in the Agent only ever selecting a plant. The policy is implemented as a heuristic decision using the shortest path to the closest plant or stag. We also deploy a human-like agent (agent B) that learns its policy from scratch using reinforcement learning. Although our approach involves multiple agents, it primarily focuses on the reinforcement learning of a single agent (Agent B) with the other Agent's behaviour predefined. Therefore, this work is rooted in a single-agent RL framework. However, it incorporates stochastic environmental influences from the other agent.

The reward function is as defined in Table 1: if agent B captures a plant (moves to a cell occupied by a plant), it receives a positive reward of $R_{plant} = 1$. If two agents capture the stag (simultaneously move to the cell occupied by the stag), they both receive a positive reward of $R_{stag} = 5$. The episode ends when either the stag or a plant is captured by agent B or a maximum number of steps T is reached. We set $T = 30$ for all experiments.

4 Theory

In this paper, we propose that visual search and target selection (stag or plant) strategies emerge as an adaptation to the environmental factors (i.e., spacing between objects; behaviour of the other agent) and the limitations of human visual and motor system (including the imprecision of peripheral vision and the inherent variability in eye movement control). We approach gaze selection as a problem of sequential decision-making, modelling it as a POMDP and using reinforcement learning to derive nearly optimal visual search and selection strategies. For an overview and broader background of this approach, see (Oulasvirta et al., 2022). In the following paragraphs, we report the theoretical assumptions.

Saccade duration Human eye movements alternate between saccades and fixations. Fixations are periods of relative eye stability, albeit with slight involuntary movements (jitter) (Duchowski, 2018), during which the eyes gather visual information. Saccades are rapid eye movements linking these fixations. For movements up to 20 degrees of visual angle, there is a consistent formula relating saccade amplitude to its duration: Duration = 2.7 × Amplitude + 37 (Baloh et al., 1975), where the saccade amplitude is in degrees, and the duration is in milliseconds (ms).

Spatial visual uncertainty Human vision is sharpest at the fovea, covering about 1 - 2 degrees of visual angle, with sharpness decreasing significantly as one moves away from this central point (Duchowski, 2018). Peripheral vision experiences marked declines in the perception of colour, shape, and size (Kieras & Hornof, 2014). Studies have shown that the variability in estimating the location of a target in peripheral vision increases linearly with its eccentricity (Michel & Geisler, 2011), a fact we incorporate into our model.

Ocular motor noise The ocular motor noise is a major source of variability in saccadic eye movements and can lead to either overshooting or undershooting of targets (Wolf & Lappe, 2021). A Gaussian noise has been previously used for modelling uncertainty in target localisation (Guadron et al., 2022; Chen et al., 2021)

5 Model

As stated above, we formalise the strategies for visual search and target selection (stag or plant) using the framework of a POMDP (Spaan, 2012). The formulation can be represented as a tuple $(S, A, O, T, Z, R, \gamma)$, where S is a set of states representing possible configurations of the environment. A is a set of actions available to the agent, such as moving towards the stag or the plant and choosing where to look next. O is a set of observations that the agent can perceive, subject to the constraints of the human visual system. $T : S \times A \rightarrow \delta(S)$ is the transition function. $Z : S \times A \rightarrow \delta(O)$ is the observation function, dictating the probability of observing each possible observation given a state and action. $R : S \times A \rightarrow R$ is the reward function, specifying the reward received after taking an action in a given state. γ is the discount factor, representing the difference in importance between future rewards and immediate rewards. The agent's objective is to learn a policy $\pi(a_t|o_{1:t})$ at each step t. The policy is represented by a recurrent neural network (RNN), which maps the observation history $o_{1:t}$ to a hidden state h_t. Consequently, at each time step, the agent samples the game environment integrates the sampled information over time and makes two choices: (a) whether to move closer to the plant or the stag and (b) where to look next for the subsequent time step. A formal description of the POMDP is given below.

State: At each time step t, the environment is occupied at a state s_t, $(s_t \in S)$. A state is represented as an RGB image of the game environment of size 160x160x3 pixels (Figure 1). The image consists of two agents of the same shape but differ in colour, two plants and a stag placed in different grid cells. Within each episode, only the position of the two agents can change.

Action: An action, a_t , is taken at each time step t . On each step, the agent decides where to saccade in the grid world and where to move the agent it controls (agent B). We use an action composition, considering an action composed of some smaller independent discrete actions. Namely, a_t is composed of a set of smaller actions D = $\{a_t^{move}, a_t^{gaze}\}$, *the gaze action space* a_t^{gaze} (the space of possibilities for where to saccade next) and *the player action space* a_t^{move} (the space of possibilities for issuing a movement action for the controlled agent). The player action space is a 5-dimensional vector of discrete values for moving the agent as described in section 3. The gaze action space is a 25-dimensional vector of discrete values representing the centre of each 5x5 grid cell. Each discrete value is then mapped to the x,y coordinate where $x, y \in [-1, 1]$ with -1 and 1 being the edge of the image. The aim saccade is corrupted by the ocular motor noise. Specifically, the actual landing position after the saccade is sampled from $a_t^{gaze} \sim N(a_t^{gaze}, \sigma_{occular}(t))$. The ocular motor noise is linearly dependent on the saccadic amplitude (Harris & Wolpert, 1998): $\sigma_{occular}(t) = \rho_{occular} \times amplitude(t)$, where $\rho_{occular}$ is a hyper-parameter of the model set to $\rho_{occular} = 0.01$ (Chen et al., 2021).

Reward: At each time step t , the environment (in one of the states s_t) generates a reward $r(s_t, a_t)$, in response to the action taken a_t. The reward function used in the model is $r(s_t, a_t)$

= task reward(t)− movement cost(t)− 0.1×(saccade duration(t) + fixation duration)/1000. Where task and movement reward is defined in section 3, saccade duration in section 4 and fixation duration set to 200ms.

Observation: At each step t, the agent receives an observation of the environment in the form of an image s_t. The agent cannot access this image fully but can sample information from s_t via a glimpse sensor. We reuse the implementation as described by Mnih et al. (2014). The sensor extracts a retina-like image pyramid (Geisler & Perry, 1998) representation $g_t = f(s_t, a_{t-1}^{gaze})$ around location a_{t-1}^{gaze} from image s_t. It encodes the region around a^{gaze} at a high resolution but uses a progressively lower resolution for pixels further from a^{gaze}, resulting in a vector of much lower resolution than the original image. Another source of uncertainty in the human visual system is the localisation error (Levi, 2008), where information in the periphery may erroneously combine features from one location with adjacent locations. In the model, spatial smearing is represented by a weighting function (Gaussian blur) with $\sigma = \rho_{smear} * 2^i$ where i is the glimpse sequence and $\rho_{smear} = 0.3$. The observation space is a tuple representing the glimpse view and the location the glimpse is extracted from, $o_t = (g_t, a_{t-1}^{gaze})$.

5.1 Internal State

The state of the environment is not directly known to the model. For this reason, the agent maintains an internal representation of the environment, summarising information extracted from past observation history. The agent uses this summarised representation to control the glimpse sensor, decide where to next sample information from, and move the agent. Specifically, at each time step t, the agent takes an action a_t and observes o_{t+1}, the internal state representation is formed by the hidden units h_{t+1} of the recurrent neural network (LSTM) and updated over time.

5.2 Intent Classification Task

Auxiliary tasks have previously been adopted to facilitate representation learning (Jaderberg et al., 2016; Lin et al., 2019). In our model, we introduce an auxiliary task head to classify the intent of the policy the opponent is pursuing. The classification layer takes the summarised observed history h_t as input and outputs a softmax function to predict the opponent's intent, with the output probabilities indicating different intent classes $c \in (stag, plant)$. The model is trained using the cross-entropy loss, where $M = 2$ represents the number of classes, y is the binary indicator (0 or 1) if class label c is the correct classification given h_t, and p is the predicted probability parameterised by a parameter vector δ given h_t is of class c.

$$L^{intent} = \sum_{c=1}^{M} y_{h_t,c} log(p^{\delta}_{h_t,c})$$

5.3 Training

Network Architecture. The architecture supports dynamic interaction with a game environment by focusing on localised sensing and memory-driven decision-making optimised by rewards. Figure 1 represents an agent architecture designed for a game environment comprising several interconnected components. A *Glimpse Sensor* (Mnih et al., 2014) captures specific portions of the environment to produce glimpse patches, which are then processed by the *Encoder* that integrates glimpse patches through a convolution neural network stack and the xy location where the patches are extracted from through a linear transformation. This encoded data is stored in a *Memory* system, utilising the LSTM layer. Decision-making is further refined by a *Gaze Controller*, which directs the agent's attention, and the *Movement Controller* dictates the agent's physical actions within the game based on internal rewards and stored memories. Appendix A provides each component's specific param-

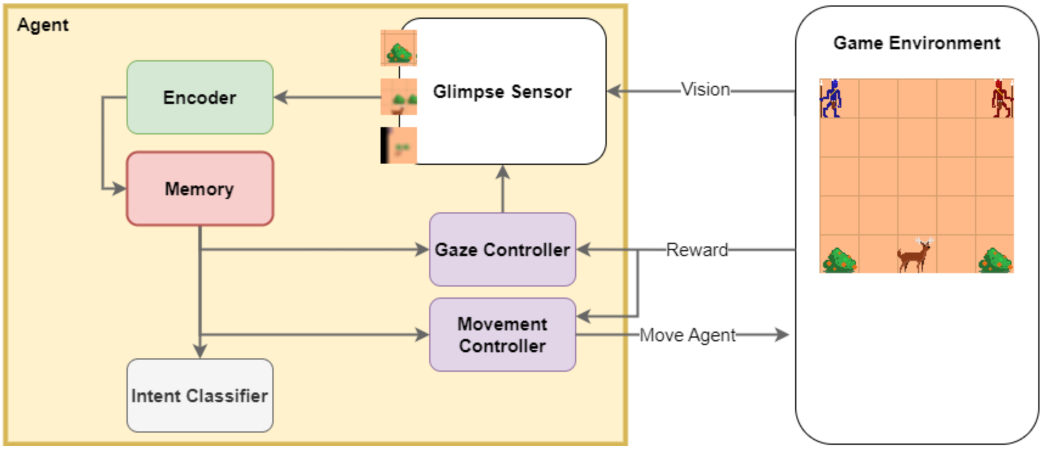


Figure 1: An overview of the Modelling approach. The control problem is modelled as deciding where to look next and where to move the agent next. The agent does not have direct access to the state of the game scenario – it must rely on selective glimpses from the glimpse sensor. There are three internal modules: the gaze controller moves the gaze and controls the gaze sensor to observe the game from pixels through foveated and peripheral vision; the movement controller moves the agent in the game based on the observations; and the memory module maintains a history of what has been seen thus far. The reward is defined as an information gathering and movement trade-off: gathering information helps achieve higher rewards but at a cost.

eters. The value network shares the *Memory* and the *Encoder* layer to learn the value of the state $V(h_t)$.

Policy Optimisation. We use the policy gradient method to train the RL agent, specifically the PPO algorithm (Schulman et al., 2017) using the CleanRL implementation (Huang et al., 2022). The goal is to learn a stochastic policy $\pi_\theta(a_t^{move}, a_t^{gaze}|o_{1:t})$, parameterised by a parameter vector θ, assigns a probability value to an action given the observation history. The model optimises the policy by maximising the expected discounted return of the policy:

$$J = \mathbb{E}_\tau \left[\sum_{t=0}^{T-1} \gamma^t r_t \right]$$

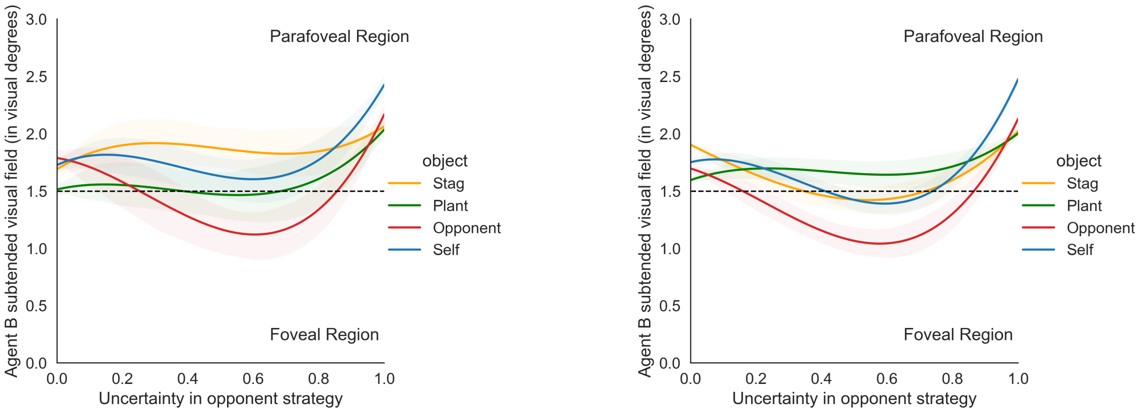
where τ is the trajectory $(s_0, a_0, r_0, o_0, ..., s_{T-1}, a_{T-1}, r_{T-1}, o_{T-1})$. The core idea behind policy gradient algorithms is to obtain the policy gradient $\nabla_\theta J$ of the expected discounted return with respect to the policy parameter vector θ. By performing a gradient optimisation step $\theta = \theta + \nabla_\theta J$, such that the expected discounted rewards are maximised. We use the following objective function to optimise:

$$\nabla_\theta J = \sum_{t=0}^{T-1} \nabla_\theta \left(\sum_{a_t^d \in D} log \pi_\theta(a_t^d|o_{1:t}) \right) A_t, A_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$$

6 Results

The results presented in this section are aggregated across 100 episodes after training for 17,000,000 time steps. The model achieved a cumulative episode reward of 2.7 (see Appendix B).

In the presented figure 2, the gaze distance (measured in visual degrees) to the Stag, Plant, Agent A (opponent), and Agent B (self) is analysed under two different pursuit policies by Agent A within



(a) Agent A Pursuit Plant Policy

(b) Agent A Pursuit Stag Policy

Figure 2: The plots show Agent B's gaze strategy as a function of the distance between its gaze position (a^{gaze}) and the game objects in visual degrees.

a game context. These policies are the Plant Policy (Figure 2a) and the Stag Policy (Figure 2b), evaluated across varying levels of uncertainty in the opponent's target pursuit policy. The figures delineate two regions based on visual angle: the foveal region (1-1.5 degrees) and the parafoveal region (1.5-3 degrees).

Plant Policy Observations: The Plant target refers to the plant closest to Agent B. The plot shows Agent B's clear strategy to consider the Plant as a significant target when there is high certainty about the opponent's strategy. As uncertainty increases, the distance to the Plant remains relatively stable, indicating that the Plant maintains its salience even with increasing uncertainty about the opponent's strategy.

Stag Policy Observations: In comparison with the plant policy, Agent B's attention is drawn towards the stag and itself under moderate uncertainty. Under high certainty about the opponents' pursuit policy, the stag policy might confirm that the plant is not salient (because the model has sufficient information from Agent A and stag, any further visual sampling is to gather additional information to confirm its policy rather than commit to a policy change).

These findings indicate that Agent A's chosen pursuit policy significantly influences Agent B's strategies, affecting how the agent interacts with other environmental elements. The adaptive nature of Agent B's gaze behaviour reflects a dynamic strategy that adjusts focus based on the certainty of the opponent's actions. The data shows that Agent B reallocates attention to the most relevant targets (Plant, Stag, Opponent, or Self) depending on the level of uncertainty, indicating a flexible and context-sensitive approach to decision-making.

7 Discussion

While the results presented here are preliminary, they hint that the agent's strategic control choices emerge as an adaptation to the constraints imposed by the human visual information processing system and the other agent's intent.

These results imply two contributions to MARL. The first is the insights into the human-machine dynamics that arise in cooperative settings, such as the effects of different pursuit policies on the agents' interactions, the trade-offs between sampling information and pursuing targets, and the role of visual attention in coordinating or disrupting joint action. Second, the design of RL agents for exploring the resource rational (Howes et al., 2009; Lieder & Griffiths, 2020) adaptation of strategies to known information processing constraints and dynamic environments. This framing is important because it helps make the crucial link between cognitive mechanism and rationality (Lewis et al., 2014) that supports deep explanations of behaviour.

One implication of these results is that the design of RL agents in multi-agent teams that could include humans should consider the human-like characteristics of their potential partners or opponents, such as their visual attention, pursuit policies, and adaptive strategies. By incorporating these factors into the agent's learning process, the agent could achieve a higher level of performance and coordination in cooperative settings and better understand the human's intentions and actions. These results show that they provide insights into the human-machine dynamics that emerge in cooperative scenarios, such as the effects of different pursuit policies on the agents' interactions, the trade-offs between sampling information and pursuing targets, and the role of visual attention in coordinating or disrupting joint action. These insights could help researchers and practitioners evaluate and improve the quality and efficiency of human-machine cooperation and identify and address potential challenges or risks. For example, the results could inform the design of feedback mechanisms, reward structures, or intervention strategies that could enhance human-machine collaboration or prevent conflicts or errors. Furthermore, the results could contribute to the theoretical understanding of human cognition and behaviour in complex and dynamic environments, such as how humans adapt to different agents and how they balance exploration and exploitation in uncertain and competitive situations.

There is a substantial amount of work to be done. First, we need to run experiments for more random seeds; second, we need to compare these results with human participants; and third, we need to systematically explore the cognitive parameter space in the model. For example, the effect of ocular motor and smearing noise on strategy, different peripheral representations for acuity decline (Lukanov et al., 2021), memory decay rate, and microsaccade within a fixation to simulate user fatigue.

Acknowledgments

The work reported in this paper was partly supported by an award from the DEVCOM Army Research Laboratory to the Alan Turing Institute and the University of Birmingham.

References

Robert W Baloh, Andrew W Sills, Warren E Kumley, and Vicente Honrubia. Quantitative measurement of saccade amplitude, duration, and velocity. *Neurology*, 25(11):1065–1065, 1975.

Nicholas J Butko and Javier R Movellan. I-pomdp: An infomax model of eye movement. In *Development and Learning, 2008. ICDL 2008. 7th IEEE International Conference on*, pp. 139–144. IEEE, 2008.

Noshaba Cheema, Laura A Frey-Law, Kourosh Naderi, Jaakko Lehtinen, Philipp Slusallek, and Perttu Hämäläinen. Predicting mid-air interaction movements and fatigue using deep reinforcement learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020.

Xiuli Chen, Aditya Acharya, Antti Oulasvirta, and Andrew Howes. An adaptive model of gaze-based selection. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pp. 1–11, 2021.

Andrew T Duchowski. Gaze-based interaction: A 30 year retrospective. *Computers & Graphics*, 73: 59–69, 2018.

Wilson S Geisler and Jeffrey S Perry. Real-time foveated multiresolution system for low-bandwidth video communication. In *Human vision and electronic imaging III*, volume 3299, pp. 294–305. SPIE, 1998.

Gerd Gigerenzer. *Adaptive thinking: Rationality in the real world.* Oxford University Press, 2002.

Leslie Guadron, A John van Opstal, and Jeroen Goossens. Speed-accuracy tradeoffs influence the main sequence of saccadic eye movements. *Scientific reports*, 12(1):5262, 2022.

Christopher M Harris and Daniel M Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394(6695):780–784, 1998.

Mary Hayhoe and Dana Ballard. Modeling task control of eye movements. *Current Biology*, 24(13): R622–R628, 2014.

Andrew Howes, Richard L Lewis, and Alonso Vera. Rational adaptation under task and processing constraints: implications for testing theories of cognition and action. *Psychological review*, 116 (4):717, 2009.

Andrew Howes, Xiuli Chen, Aditya Acharya, and Richard L Lewis. Interaction as an emergent property of a partially observable markov decision process. *Computational interaction design*, pp. 287–310, 2018.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Laurent Itti and Christof Koch. A saliency-based search mechanism for overt and covert shifts of visual attention. *Vision research*, 40(10):1489–1506, 2000.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Jussi Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. Touchscreen typing as optimal supervisory control. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pp. 1–14, 2021.

David E Kieras and Anthony J Hornof. Towards accurate and practical predictive models of active-vision-based visual search. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 3875–3884, 2014.

Dennis M Levi. Crowding an essential bottleneck for object recognition: A mini-review. *Vision research*, 48(5):635–654, 2008.

Richard L Lewis, Andrew Howes, and Satinder Singh. Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in cognitive science*, 6(2): 279–311, 2014.

Falk Lieder and Thomas L Griffiths. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and brain sciences*, 43:e1, 2020.

Xingyu Lin, Harjatin Baweja, George Kantor, and David Held. Adaptive auxiliary task weighting for reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Hristofor Lukanov, Peter König, and Gordon Pipa. Biologically inspired deep learning model for efficient foveal-peripheral vision. *Frontiers in Computational Neuroscience*, 15:746204, 2021.

and Demir Mustafa McNeese, Nathan J., Nancy J. Cooke, and Christopher Myers. Teaming with a synthetic teammate: Insights into human-autonomy teaming. *Human Factors*, 60(2):262–274, 2018.

Melchi Michel and Wilson S Geisler. Intrinsic position uncertainty explains detection and localization performance in peripheral vision. *Journal of Vision*, 11(1):18–18, 2011.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. Recurrent models of visual attention. *Advances in neural information processing systems*, 27, 2014.

Syed Kumayl Raza Moosavi, Muhammad Hamza Zafar, and Filippo Sanfilippo. Collaborative robots (cobots) for disaster risk resilience: a framework for swarm of snake robots in delivering first aid in emergency situations. *Frontiers in Robotics and AI*, 11:1362294, 2024.

Christopher W Myers, Richard L Lewis, and Andrew Howes. Bounded optimal state estimation and control in visual search: Explaining distractor ratio effects. In *Proc. CogSci*, 2013.

Jiri Najemnik and Wilson S Geisler. Eye movement statistics in humans are consistent with an optimal search strategy. *Journal of Vision*, 8(3):4–4, 2008.

David Lvovich Nesterov-Rappoport. The evolution of trust: Understanding prosocial behavior in multi-agent reinforcement learning systems. Technical report, Drew University, Madison, NJ, May 2022.

Jose Nunez-Varela and Jeremy L Wyatt. Models of gaze control for manipulation tasks. *ACM Transactions on Applied Perception (TAP)*, 10(4):20, 2013.

Antti Oulasvirta, Jussi PP Jokinen, and Andrew Howes. Computational rationality as a theory of interaction. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2022.

Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pp. 2043–2044, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

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

Matthijs T. J. Spaan. Partially observable markov decision processes. In Marco Wiering and Martijn van Otterlo (eds.), *Reinforcement Learning: State of the Art*, pp. 387–414. Springer, 2012.

Christian Wolf and Markus Lappe. Vision as oculomotor reward: cognitive contributions to the dynamic control of saccadic eye movements. *Cognitive Neurodynamics*, 15(4):547–568, 2021.

Jeremy M Wolfe. Guided search 4.0. *Integrated models of cognitive systems*, pp. 99–119, 2007.

Chen Xiuli, Sandra Dorothee Starke, Chris Baber, and Andrew Howes. A cognitive model of how people make decisions through interaction with visual displays. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pp. 1205–1216. ACM, 2017.

A Model Hyperparameters

A.1 Hyperparameters used for training the policy

Parameter Names	Parameter Values
N_{total} Total Time Steps	17000000
N_{mb} Number of Mini-Batches	4
N_{envs} Number of Environments	4
N_{steps} Number of Steps per Environment	256
γ Discount Factor	0.99
λ for GAE	0.95
ϵ PPO clipping coefficient	0.2
Maximum Gradient Norm	0.5
K Number of updates per epoch	4
α Learning Rate	0.00025 linearly decreased to 0 over total time steps
Value Function Coefficient	0.5
Entropy Coefficient	0.01
$N_{updates}$ Total Updates	$N_{total}/N_{mb}N_{envs}$

A.2 Hyperparameters used for Glimpse module

Parameter Names	Parameter Values
Glimpse Size	40 pixels
Number of Patches	3
Patch Scale	2

A.3 Hyperparameters used for Encoder module

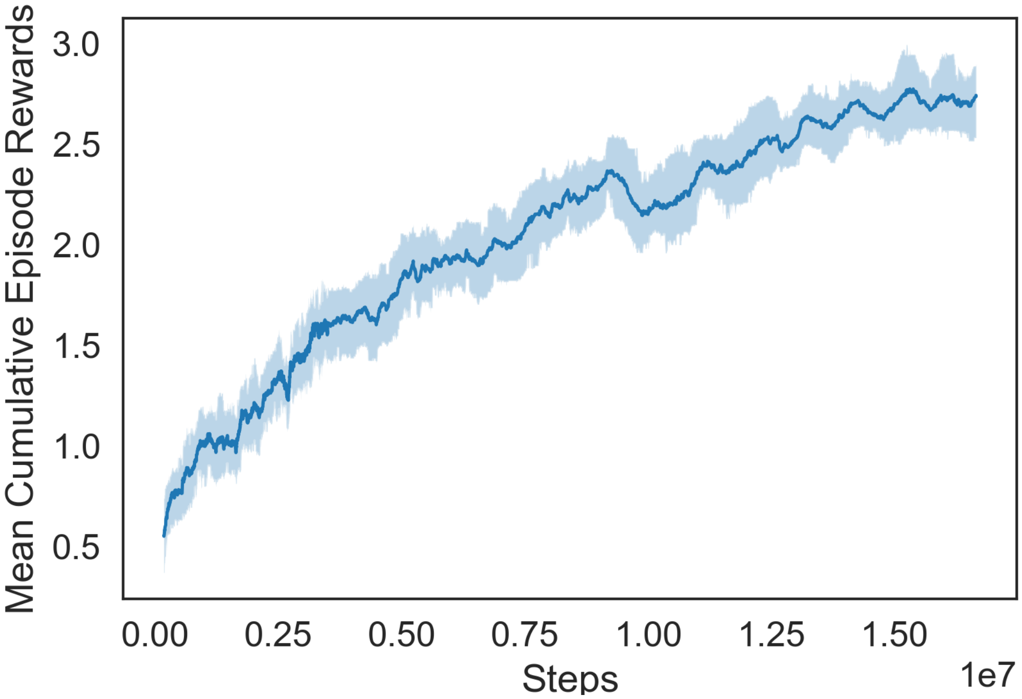
Parameter Names	Parameter Values
Number of CNN Layers	3
Output channels	[16, 32, 64]
Kernel Size	[4, 4, 3]
Stride	[4, 2, 1]
Activation function	Relu
Linear Layer size	32

A.4 Hyperparameters used for Controller and Memory module

Parameter Names	Parameter Values
Controller Number of Linear Layers	2
Controller Hidden Layers size	32
Gaze controller Output Layers size	25
Movement controller Output Layers size	5
Activation function	Relu
Memory LSTM Size	64

Layers were initialised using orthogonal initialisation. Final action layers was initialised with standard deviation 0.01 and value network with standard deviation 1. Bais was initialised with a constant 0.

B Model Convergence



(a) mean cumulative episode rewards with IQR