Narrowing Action Choices with AI Improves Human Sequential Decisions

Eleni Straitouri* Max Planck Institute for Software Systems

Stratis Tsirtsis* Hasso Plattner Institute

Ander Artola Velasco Max Planck Institute for Software Systems

Manuel Gomez Rodriguez Max Planck Institute for Software Systems

Abstract

Recent work has shown that, in classification tasks, it is possible to design decision support systems that do not require human experts to understand when to cede agency to a classifier or when to exercise their own agency to achieve complementarity—experts using these systems make more accurate predictions than those made by the experts or the classifier alone. The key principle underpinning these systems reduces to adaptively controlling the level of human agency, by design. Can we use the same principle to achieve complementarity in sequential decision making tasks? In this paper, we answer this question affirmatively. We develop a decision support system that uses a pre-trained AI agent to narrow down the set of actions a human can take to a subset, and then asks the human to take an action from this action set. Along the way, we also introduce a bandit algorithm that leverages the smoothness properties of the action sets provided by our system to efficiently optimize the level of human agency. To evaluate our decision support system, we conduct a large-scale human subject study ($n = 1{,}600$) where participants play a wildfire mitigation game. We find that participants who play the game supported by our system outperform those who play on their own by $\sim$30% and the AI agent used by our system by $>$2%, even though the AI agent largely outperforms participants playing without support.

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300. Copyright 2026 by the author(s). *Equal contribution.

1 INTRODUCTION

The idea that humans and machines can work together to achieve greater outcomes than what they can achieve on their own—in short, complementarity—has long intrigued the research community (Jordan, 1963). Long before the development of modern machine learning, several lines of work provided early evidence that complementarity can be more than just an intriguing idea and may, in fact, be achievable in practice (Brodley et al., 1999; Horvitz and Paek, 2007). Since then, a flurry of work has been growing this evidence, especially in the context of automated decision support systems for one-shot prediction tasks (Kamar et al., 2012; Raghu et al., 2019; Choudhury et al., 2020; De et al., 2020; Bengs and Hüllermeier, 2020; De et al., 2021; Mozannar and Sontag, 2020; Okati et al., 2021; Kerrigan et al., 2021; Bansal et al., 2021a; Steyvers et al., 2022; Arnaiz-Rodriguez et al., 2025).

The conventional wisdom is that, to achieve complementarity in such tasks, human decision makers need to understand when to cede agency to a predictor (*e.g.*, a classifier) or when to exercise their own agency (Yin et al., 2019; Zhang et al., 2020; Suresh et al., 2020; Lai et al., 2021; Bansal et al., 2021b; Corvelo Benz and Gomez-Rodriguez, 2023, 2025). However, a very recent line of work has introduced an alternative family of decision support systems for classification tasks that, perhaps surprisingly, achieve complementarity without such a (strong) requirement (Straitouri et al., 2023; Straitouri and Gomez-Rodriguez, 2024; Toni et al., 2024). The key principle underpinning these systems reduces to adaptively controlling the level of human agency—rather than providing a single prediction and letting a human decide when and how to use the prediction, these systems provide a set of predictions, and ask the human to pick one prediction from the set. In this work, we investigate whether the same principle can be applied to achieve complementarity in sequential decision making tasks.

Our contributions. We develop a decision support system for sequential decision making tasks that uses a pre-trained AI agent to narrow down the set of actions a human decision maker can take to a subset, namely an *action set*, and asks the human to take an action from the action set. To develop this system, we start by characterizing how humans make sequential decisions with action sets as a structural causal model (SCM) (Pearl, 2009) where, over time, a human collects rewards based on their chosen actions and their resulting effects on the environment's state. Building upon this characterization, we show that our decision support system constructs action sets in a way such that the average total reward achieved by a human using our system is Lipschitz continuous with respect to a parameter controlling the size of the action sets and, consequently, the level of human agency. Then, we leverage this property to develop an efficient Lipschitz bandit algorithm with sublinear regret guarantees that identifies the optimal level of agency in terms of average total reward, which may be of independent interest as discussed in the comparison with further related work.

Finally, we evaluate our decision support system by conducting a large-scale human subject study, where 1,600 participants play 16,000 instances of a wildfire mitigation game. We find that participants who play the game supported by our system outperform those who play on their own by $\sim$30% and the AI agent used by our system by $>$2%, even though the AI agent used by our system largely outperforms participants playing without support. We have made available the data gathered in our human subject study as well as an open source implementation of our system at `https://github.com/Human-Centric-Machine-Learning/narrowing-action-choices`.

Further related work. Our work builds upon further related work on decision support in sequential decision making, multi-armed bandits, and action masking in reinforcement learning.

Within the area of decision support in sequential decision making, the work most closely related to ours provides humans with advice while taking actions. The advice can be in the form of "tips" generated by reinforcement learning (Bastani et al., 2025) or in the form of action recommendations (Grand-Clément and Pauphilet, 2024). However, in contrast to our work, humans have the option to ignore the advice and exercise their own agency. Within this area of research, our work also relates to the notion of algorithmic triage (Straitouri et al., 2021; Zhang et al., 2021; Balazadeh et al., 2022), where a human has to take actions for a fraction of the environment states, and an AI agent takes actions for the rest.

Within the area of multi-armed bandits, our work closely relates to the literature of Lipschitz bandits (Agrawal, 1995; Kleinberg et al., 2008; Bubeck et al., 2008)—refer to Slivkins et al. (2019) for a detailed review on the large family of multi-armed bandit algorithms. In our work, we view the parameter controlling the level of human agency as an arm of a continuum-armed bandit problem, and show that the average total reward is a Lipschitz function of the arm. Then, we leverage this property to develop a Lipschitz bandit algorithm for best arm identification with simple regret guarantees. Our algorithm can be of independent interest, as it is the first to derive simple regret guarantees with no assumptions on the error of the finite sample estimate of the expected payoff of each arm. Existing work on Lipschitz best arm identification with simple regret guarantees (Feng et al., 2023) makes rather explicit assumptions on the convergence rate of this error.

Action masking is a technique used by reinforcement learning algorithms that filters—masks out—invalid actions based on the state of the environment (*e.g.*, unavailable actions given the rules of a game) before sampling an action (Vinyals et al., 2017; Berner et al., 2019; Ye et al., 2020; Huang and Ontañón, 2022; Qian et al., 2022; Müller and Sabatelli, 2022; Wang et al., 2024). Action masks can be manually designed based on domain knowledge or learned explicitly using parameterized models, and their goal is to efficiently guide exploration in reinforcement learning algorithms by excluding actions that are irrelevant, unavailable or unsafe given a specific environment. However, action masks are fundamentally different to our action sets since they focus on increasing the total reward achieved by algorithms rather than humans.

2 SEQUENTIAL DECISION MAKING WITH ACTION SETS

In this section, we first characterize how humans make sequential decisions using our decision support system based on action sets via a structural causal model (SCM) (Pearl, 2009), building upon previous work on sequential decision making (Oberst and Sontag, 2019; Tsirtsis et al., 2021) and decision support systems based on prediction sets (Straitouri and Gomez-Rodriguez, 2024).[1] Then, we describe how our decision support system specifically constructs action sets using state-of-the-art AI agents.

At each time step $t \in \mathbb{N}$, a human decision maker observes the state of the environment $\boldsymbol{Z}_t \in \mathcal{Z}$, and our

[1]We use upper case letters for random variables, lower case letters for their realizations, calligraphic letters for sets, and bold letters for vectors, unless stated otherwise.

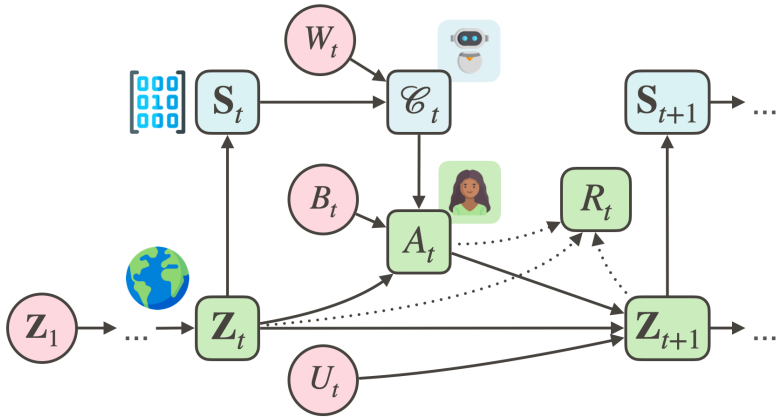


Figure 1: **A causal model of human sequential decision making with action sets.** Squares represent endogenous random variables and circles represent exogenous noise variables. An arrow from (to) one variable indicates that it is the cause (effect) of another. The values of all exogenous variables are sampled from fixed (unknown) distributions, while the value of each endogenous variable is given by a function of the values of its ancestors in the causal graph.

decision support system observes a (potentially lossy) representation $\boldsymbol{S}_t = f_S(\boldsymbol{Z}_t) \in \mathcal{S}$. This captures a variety of real-world scenarios in which the human decision maker has access to salient but hard-to-quantify information that makes complementarity a particularly worthy goal (Alur et al., 2023); for example, a doctor's direct conversations with patients in healthcare, a scout's qualitative assessment of prospective players in professional sports, or an officer's handling of resource constraints in disaster response. Then, given a finite space of m possible actions $\mathcal{A}$, our decision support system narrows down the space of actions the human can take to an action set

$$\mathcal{C}_t = \pi_{\mathcal{C}}(\boldsymbol{S}_t, W_t) \subseteq \mathcal{A}, \tag{1}$$

where $\pi_{\mathcal{C}}$ is a decision support policy, and $W_t \sim P(W)$ is an (exogenous) noise variable that allows for randomized action sets.[2] Further, given the action set $\mathcal{C}_t$ provided by our decision support system and the current state $\boldsymbol{Z}_t$ of the environment, the human takes an action

$$A_t = \pi_H(\boldsymbol{Z}_t, \mathcal{C}_t, B_t), \tag{2}$$

where π_H is an (unknown) human policy, and B_t is a noise variable that captures randomness in human decisions. Subsequently, depending on the action A_t taken by the human and the current state $\boldsymbol{Z}_t$, the environment transitions to the next state $\boldsymbol{Z}_{t+1} = f_Z(\boldsymbol{Z}_t, A_t, U_t)$, where U_t is a noise variable that captures inherent randomness in the environment's state transitions.[3] Finally, following each transition, the human receives a numerical reward $R_t = f_R(\boldsymbol{Z}_t, A_t, \boldsymbol{Z}_{t+1}) \in [-r_{\max}, r_{\max}]$. Figure 1 shows a visual representation of our causal model.

To define the decision support policy $\pi_{\mathcal{C}}$ and in turn construct action sets $\mathcal{C}_t$, our decision support system utilizes an AI agent's valuation $q(\mathbf{s}, a)$ of each action $a \in \mathcal{A}$ given a representation $\mathbf{s} \in \mathcal{S}$. The AI agent's valuation may correspond, for example, to Q-values given by a deep Q-network in the context of reinforcement learning (Mnih et al., 2013) or logits over a set of possible actions given by a large language model fine-tuned for sequential decision making (Li et al., 2022).[4] More specifically, let $a_{(1)}, a_{(2)}, ..., a_{(m)}$ be the ranking of the actions $a \in \mathcal{A}$ according to the AI agent's valuation $q(\mathbf{s}, a)$. Then, the decision support policy $\pi_{\mathcal{C}}$ is given by

$$\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \{a_{(i)}\}_{i=1}^{k}, \text{ where}$$
$$k = 1 + \sum_{j=2}^{j=m} \mathbb{1}\left\{\tilde{q}(\mathbf{s}, a_{(j)}) + W \geq \tilde{q}(\mathbf{s}, a_{(1)}) - \varepsilon\right\}, \tag{3}$$

and W is a half-normal noise variable $W = |X|$ with $X \sim \mathcal{N}(0, \sigma^2)$, $\tilde{q}(\mathbf{s}, a) \in [0, 1]$ denotes a scaled version of the original valuation $q(\mathbf{s}, a)$ computed via min-max normalization, and $\varepsilon \in [0, 1]$ is a parameter controlling the size of the action set and, consequently, the level of human agency.

[2] In Section 3, we leverage this randomization to optimize the decision support policy $\pi_{\mathcal{C}}$ using Lipschitz bandit algorithms.

[3] Note that the state $\boldsymbol{Z}_t$ may include the entire history of past states and actions, and the noise variables $\boldsymbol{U}_t$ may be correlated across time steps. As a consequence, our model allows for both Markovian and non-Markovian environment dynamics and human policies.

[4] Whenever AI agents operate autonomously, they often implement a greedy policy $\pi_M(\mathbf{s}) = \text{argmax}_{a \in \mathcal{A}}\, q(\mathbf{s}, a)$. However, if $\mathbf{s}$ is a lossy representation of $\mathbf{z}$, such a greedy policy is likely to be suboptimal.

Intuitively, the above decision support policy gives a small random boost W to the (scaled) valuations of all actions except for the best one, and it includes in the action set all actions whose boosted valuations are within ε of the valuation of the best action. Our motivation for this construction is twofold. First, it allows us to control the level of human agency using a (continuous) parameter that is independent of the size of the action space $\mathcal{A}$. For example, for $\varepsilon = 1$, the action set contains all actions $\mathcal{A}$ and thus the human maintains full agency and, for $\varepsilon = 0$ and σ sufficiently small, the action set contains only the highest ranked action $a_{(1)}$ and the human has no agency. Second, the randomization introduced by the random variable W ensures that, as we will show later on, the probability distribution over candidate action sets entailed by Eq. 3 varies smoothly with ε. In what follows, we leverage this smoothness property to develop a bandit algorithm that efficiently finds the optimal value of the parameter ε controlling the level of human agency that maximizes the (discounted) cumulative reward, *i.e.*,

$$\varepsilon^* = \operatorname*{argmax}_{\varepsilon \in [0,1]} \mathbb{E}_{\boldsymbol{Z}_1, \boldsymbol{W}, \boldsymbol{B}, \boldsymbol{U}} \left[\sum_{t=0}^{+\infty} \gamma^t \cdot R_{t+1} \right], \qquad (4)$$

where $\gamma \in (0, 1)$ is a given discount factor, and the expectation is over the initial state of the decision making process Z_1 and the randomness in the decision support policy, the human's policy, and the environment's state transitions.

3 A LIPSCHITZ BANDIT ALGORITHM TO IDENTIFY OPTIMAL ACTION SETS

Our starting point is the observation that, based on Eq. 3, a decision support policy can assign non-zero probability only to m different action sets—those of the form $\mathcal{C}_{(i)} = \{a_{(1)}, a_{(2)}, \ldots, a_{(i)}\}$, where $a_{(1)}, a_{(2)}, \ldots, a_{(m)}$ is the ranking of the m actions based on the valuations $q(\mathbf{s}, a)$ of the AI agent.

Building upon that observation, we first show that the total variation distance between two distributions over the m action sets induced by two different values of the parameter ε is bounded (up to a multiplicative factor) by the absolute difference between those two values.[5]

Proposition 1 *Let $L_c = \frac{2\sqrt{2}}{\sigma\sqrt{\pi}}$. For any representation $\mathbf{s} \in \mathcal{S}$ and any $\varepsilon, \varepsilon' \in [0, 1]$, it holds that*

$$\sum_{i=1}^{m} \Big| \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] - \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \Big| \leq L_c |\varepsilon - \varepsilon'|. \qquad (5)$$

[5]The proofs for all propositions can be found in Appendix A.

Algorithm 1 Lipschitz Best Arm Identification

Input: Exploration budget n, Lipschitz constant L, exploitation parameter β.
$t_1 \leftarrow 0$, $k \leftarrow 1$, $\mathcal{I}_1 \leftarrow \{[0, 1/2], [1/2, 1]\}$
while $t_k \leq n$ **do**
  $l_k \leftarrow 2^{-k}$
  $n_k \leftarrow 2^{k\beta}$
  for $I \in \mathcal{I}_k$ **do**
    $\varepsilon_I \leftarrow \text{midpoint}(I)$
    $\hat{p}(\varepsilon_I) \leftarrow \textsc{PullArm}(\varepsilon_I, n_k)$ // Pull the midpoint of each interval n_k times
  end
  $\hat{p}_{\max} \leftarrow \max_{I \in \mathcal{I}_k} \hat{p}(\varepsilon_I)$
  $\varepsilon_{\text{OPT}} \leftarrow \textsc{Midpoint}\left(\text{argmax}_{I \in \mathcal{I}_k} \hat{p}(\varepsilon_I)\right)$
  // Find the interval whose midpoint achieves the highest average payoff
  $\mathcal{I}_{k+1} \leftarrow \varnothing$
  for $I \in \mathcal{I}_k$ **do**
    // Zoom-in and explore further only intervals whose average payoff is close to the highest
    if $\hat{p}_{max} - \hat{p}(\varepsilon_I) \leq (2 + L/2) l_k$ **then**
      $I_{\text{low}}, I_{\text{high}} \leftarrow \textsc{PartitionInHalf}(I)$
      $\mathcal{I}_{k+1} \leftarrow \mathcal{I}_{k+1} \cup \{I_{\text{low}}, I_{\text{high}}\}$
    end
  end
  $t_{k+1} \leftarrow t_k + 2|\mathcal{I}_{k+1}| n_k$
  $k \leftarrow k + 1$
end
return ε_{OPT}

Perhaps surprisingly, the above proposition suggests that the amount by which the two distributions can differ depends only on the standard deviation of the half-normal random variable W used in our decision support policy and not on the number of actions m. Further, we leverage the proposition to show that the (discounted) cumulative reward achieved by a human using our decision support policy varies smoothly with respect to the parameter ϵ. More formally, let

$$v(\mathbf{z}; \varepsilon) = \mathbb{E}_{\boldsymbol{W}, \boldsymbol{B}, \boldsymbol{U}} \left[\sum_{t=0}^{+\infty} \gamma^t \cdot R_{t+1} \mid \boldsymbol{Z}_1 = \mathbf{z} \right]$$

denote the (discounted) cumulative reward achieved by a human using a decision support policy $\pi_{\mathcal{C}}$ parameterized by ε, conditioned on an initial state $\mathbf{z}$. Then, the following proposition suggests that its expectation over the initial state $\boldsymbol{Z}_1$ is Lipschitz-continuous with respect to ε.

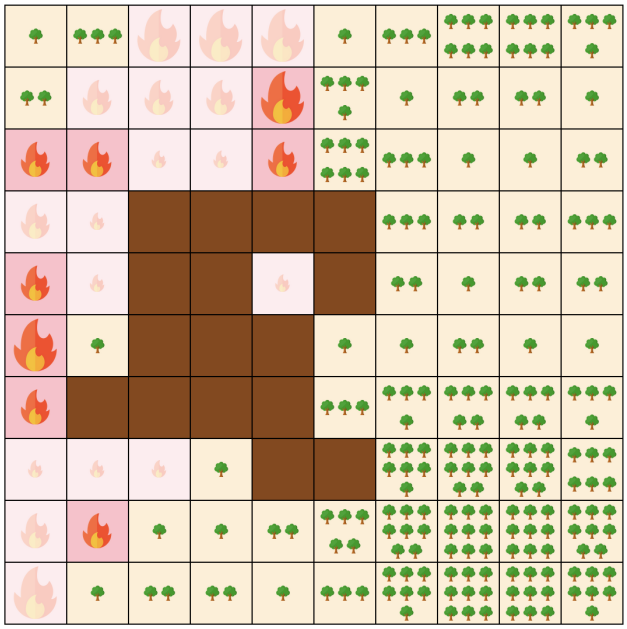


Figure 2: **Instance of our wildfire mitigation game as shown to human participants.** Tiles with green trees are `healthy`, tiles with fire icons are `burning`, and brown tiles are `burnt`. The size of the fire icon in each `burning` tile is proportional to the time steps left until the tile turns to `burnt` on its own. Non-faded `burning` tiles correspond to the action set given by our decision support system, while faded tiles correspond to actions that the decision support system does not allow the human participant to select.

Proposition 2 *Let* $L_v = \frac{L_c \cdot r_{max}}{(1-\gamma)^2}$. *For any* $\varepsilon, \varepsilon' \in [0,1]$, *it holds that*

$$|\mathbb{E}_{\boldsymbol{Z}_1}[v(\boldsymbol{Z}_1;\varepsilon)] - \mathbb{E}_{\boldsymbol{Z}_1}[v(\boldsymbol{Z}_1;\varepsilon')]| \leq L_v \cdot |\varepsilon - \varepsilon'|. \quad (6)$$

As an immediate consequence, we can find the optimal value of the parameter ε^* defined in Eq. 4 by viewing each parameter value $\varepsilon \in [0,1]$ as an arm in a best-arm identification Lipschitz bandit problem (Agrawal, 1995; Kleinberg et al., 2008; Bubeck et al., 2008). In a typical best-arm identification problem, one pulls different arms repeatedly and observes a (sampled) payoff for each pull. Based on these observations, the objective is to identify the arm that yields the maximum expected payoff. In our setting, we view the act of pulling an arm ε as a single instance of our sequential decision making task, in which the human takes a sequence of actions constrained by the action sets generated by our decision support policy with the parameter ε. Further, the observed payoff for each pull is the cumulative (discounted) reward achieved by the human in the respective instance, leading to an expected payoff for an arm ε equal to $p(\varepsilon) = \mathbb{E}_{\boldsymbol{Z}_1}[v(\boldsymbol{Z}_1;\varepsilon)]$.[6] In what follows, we develop a best-arm identification algorithm that pulls arms efficiently by leveraging the Lipschitz-continuity of the expected payoffs shown in Proposition 2.

Our algorithm, summarized in Algorithm 1, proceeds in iterations and it uses a technique known as *zooming* (Kleinberg et al., 2008). Specifically, in each iteration k, it starts with a set $\mathcal{I}_k$ of *active* (sub-)intervals of $[0,1]$, that is, intervals that are more likely to include the best arm ε^*. Then, it explores each active interval $I \in \mathcal{I}_k$ by pulling n_k times an arm ε_I representative of the interval I (*e.g.*, its midpoint) and keeping track of its empirical average payoff $\hat{p}(\varepsilon_I)$. Based on a given Lipschitz constant L for the expected payoffs, the algorithm continues by identifying and eliminating intervals that are unlikely to include the best arm. Lastly, it partitions the remaining intervals in half to construct the new set of active intervals to be explored in the next iteration. The algorithm terminates once it consumes a user-specified exploration budget of n arm pulls.

The aforementioned procedure allows us to derive guarantees on the performance of our algorithm in terms of its *simple regret*, that is, the difference between the expected payoff of the optimal arm ε^* and the arm ε_{ALG} returned by our algorithm. Importantly, the guarantees we derive depend on an instance-specific number $d \in \mathbb{N}$, known as the *zooming dimension* (Kleinberg et al., 2008), which captures how "difficult" it is to identify the best arm in a given Lipschitz bandit problem instance—low values of d indicate that arms with close-to-optimal expected payoffs are concentrated around the best arm, while high values of d indicate that such arms are spread out in the $[0,1]$ interval.[7] Formally, we have the following proposition regarding the simple regret of Algorithm 1:

Proposition 3 *Let* d *be the zooming dimension of an instance of the problem given by Eq. 4. For any* $n \geq 1$, *there exists a* $\delta_n > 0$ *such that, with probability at least* $1 - \delta_n$, *Algorithm 1 with exploitation parameter* β *returns an arm* ε_{ALG} *that satisfies*

$$p(\varepsilon^*) - p(\varepsilon_{ALG}) \leq \mathcal{O}(n^{\frac{-1}{d+\beta}}).$$

The inequality above tells us that the simple regret achieved by our algorithm improves as the exploration

[6] In practice, to be able to observe the payoff for each pull, the environment transitions need to be such that the decision making process eventually terminates, that is, it reaches an absorbing state that yields zero reward.

[7] Formally, the zooming dimension of a bandit problem is the smallest $d > 0$ such that there exists a constant $\lambda > 0$ with the following property: for every $\rho > 0$, there exist $N \leq \lambda \rho^{-d}$ intervals $I \subseteq [0,1]$ of length ρ that cover the set of arms $\{\varepsilon \in [0,1] : p(\varepsilon^*) - p(\varepsilon) \leq \rho\}$.

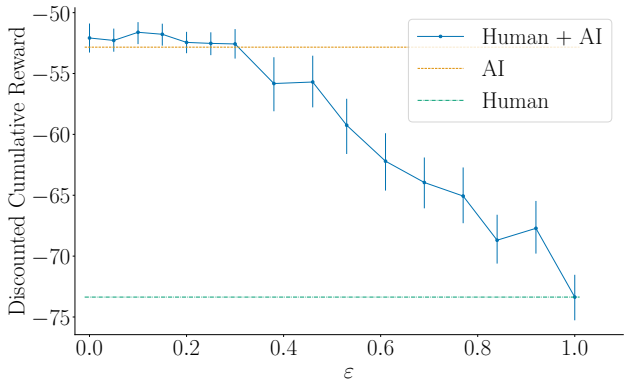


Figure 3: **Discounted cumulative reward achieved by human participants supported by our system, human participants on their own, and the AI agent used by our system.** In the blue line, each point correspond to a different value of the parameter ε controlling the level of human agency and error bars correspond to 95% confidence intervals. Here, we use $\gamma = 0.99$ and, to improve visibility, in the range of $\varepsilon \in [0, 0.3]$, we create coarse-grained bins of ε of size 5 and, for each bin, we show the average discounted cumulative reward.

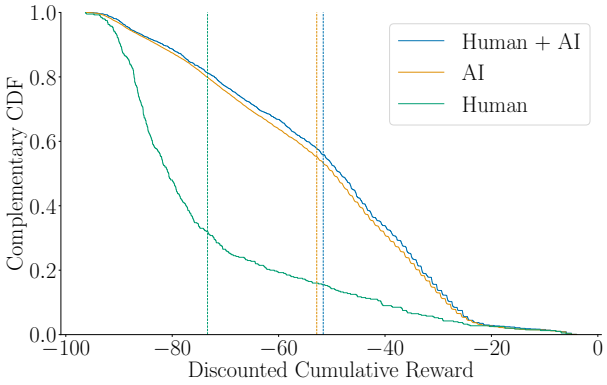


Figure 4: **Distribution of discounted cumulative rewards achieved by human participants supported by our system, human participants on their own, and the AI agent used by our system.** Within the plot, vertical lines correspond to the respective empirical average discounted cumulative reward. Here, our decision support system uses $\varepsilon \in [0.08, 0.12]$, and a Welch's t-test confirms that the difference between its cumulative reward distribution and that of the AI agent alone and the humans alone is statistically significant with $p \leq 0.01$ in both cases.

budget n increases. Moreover, the rate at which this happens depends on the zooming dimension d, with lower values of d (*i.e.*, easier instances) leading to faster rates of decrease in the simple regret. Although we have omitted the constants in the expression above for brevity, note that the simple regret also improves with lower values of the given Lipschitz constant L (refer to Eq. 11 in Appendix A.3 for the full expression). In what follows, we evaluate the performance of Algorithm 1 in efficiently identifying the optimal value of the parameter ε in a human subject study, along with the overall ability of our decision support policy introduced in Section 2 to improve human sequential decisions.

4 EVALUATION VIA HUMAN SUBJECT STUDY

In this section, we develop a wildfire mitigation game and use it to conduct a large-scale human subject study. Our goal is to show that: i) under the optimal value of the parameter ε controlling the level of human agency, our system achieves human-AI complementarity and, ii) our Lipschitz best arm identification algorithm successfully identifies the optimal value of ε, while achieving a notable improvement in simple regret compared with a competitive baseline.[8]

[8]We ran our experiments on a Debian machine with an ARM EPYC 7662 processor, 20 cores and 32GB memory.

Human subject study setup. We recruited 1,600 participants through Prolific[9]. We compensated each participant with 9 GBP per hour pro-rated, as recommended by Prolific, and we doubled the compensation of the top 10% of participants in terms of their score in our game to further incentivize good performance. Each participant provided their consent to participate by filling in a form with a detailed description of the purpose and process of the study. Our experimental protocol received approval by the Institutional Review Board (IRB) from the University of Saarland.

Each participant went through a short tutorial and then played 10 instances of our wildfire mitigation game. In each game instance, we collected data at each individual time step that included the state of the game, the action that the participant took, and the immediate (in-game) reward they received. We did not collect personally identifiable information and stored the data mentioned above in an anonymized way. For further details regarding our human subject study setup, refer to Appendix B.

Wildfire mitigation game. Each game instance includes a fictional forest map with an on-going wildfire, and the participant's goal is to prevent it from spreading. Figure 2 shows a snapshot of one of the game instances played by a participant.

The *state* of the game is characterized by the status of

[9]https://www.prolific.com/

the fictional forest, shown as a 10x10 grid, where each grid tile is either `healthy` (has green trees), `burning` (on fire), or `burnt` (brown). The number of trees in each `healthy` tile indicates how easy it is for the tile to catch fire (*i.e.*, more trees indicate an easier spread). Moreover, the size of the fire in each `burning` tile indicates how long the fire in that tile can keep burning (*i.e.*, smaller fires will extinguish themselves sooner).

At each time step of the game, our system creates an *action set* that contains a subset of the `burning` tiles (non-faded) and the participant takes an *action* by selecting one of those tiles to which they can apply water mitigation measures; these extinguish the fire and turn the respective tile to `burnt`.

The *transitions* between states and the immediate *rewards* are determined by the actions of the participant and the dynamics of the fire spread. At each time step, each `burning` tile spreads the fire stochastically to a subset of its neighboring `healthy` tiles (refer to Appendix B for details), and the immediate reward equals the negative total number of tiles that caught fire in that time step. In addition, a `burning` tile to which no water mitigation measures have been applied turns to `burnt` on its own after three time steps. Lastly, once a tile becomes `burnt` it cannot become neither `burning` nor `healthy` again. The game ends when there are no more `burning` tiles left, and the participant gets a score equal to the number of remaining `healthy` tiles.[10]

Decision support policies. We implement the decision support policies underpinning our system, defined by Eq. 3, using a range of parameter values ε from 0.0 to 0.3 with step 0.01, and from 0.3 to 1.0 with step 0.05. For each ε value, we create a set of 400 game instances in which participants play the wildfire mitigation game supported by our decision support system, and each participant is randomly assigned 10 instances with multiple ε values. As an AI agent, we train and use a Deep Q-Network (DQN) (Mnih et al., 2015) that, at each time step of a game instance, returns valuations $q(\mathbf{s}, a)$ for all actions a (*i.e.*, `burning` tiles) depending on the current representation $\mathbf{s}$ (*i.e.*, the state of the game). Lastly, for the half-normal distribution of the random variable W that adds stochasticity to the construction of action sets, we use $\sigma = 0.01$. For further details on the implementation of our decision support policies as well as the training of our Deep Q-Network refer to Appendix B.

[10]We use $\gamma \approx 1$, hence, the (discounted) cumulative reward is approximately equal to $score - 100$. This is because the immediate reward at time step t is $R_t = -n_t$, where n_t is the number of tiles that catch fire at time step t. As a result, for $\gamma \approx 1$, the (discounted) cumulative reward is $\sum_{t=0}^{\infty} \gamma^t R_{t+1} \approx -\sum_{t=0}^{\infty} n_{t+1} = \#\texttt{healthy}$ tiles $- 100 = score - 100$.

Results. Figure 3 provides evidence that our decision support system can achieve complementarity under several values of the parameter ε controlling the level of human agency. In particular, under the optimal value of the parameter ε, a human using our decision support system achieves an average (discounted) cumulative reward 29.65% higher than the one a human achieves by playing the game on their own (*i.e.*, under $\varepsilon = 1$). Perhaps surprisingly, even though the AI agent used by our system outperforms human participants playing on their own by a large margin, our decision support system under the optimal value of ε manages to further improve its average (discounted) cumulative reward by 2.31%. Figure 4 complements these findings by showing that the empirical cumulative distribution of the (discounted) cumulative reward under the optimal value of ε not only differs from the distributions of the (discounted) cumulative reward achieved by the AI agent and humans playing on their own, but also dominates them. To further support this claim, we run a Welch's t-test, which confirms that the differences between these distributions are statistically significant ($p = 0.01$ and $p = 3 \cdot 10^{-69}$, respectively) and further analyze the effect sizes (Sullivan and Feinn, 2012). Specifically, we find a large positive effect on the cumulative reward achieved by humans using our decision support system compared with humans playing on their own (Cohen's $d = 1.07$), and a small positive effect compared with the AI agent playing on its own ($d = 0.06$), suggesting that our system substantially improves human performance while also achieving a moderate form of human-AI complementarity. In Appendix C, we further support this finding through an analysis on the size of action sets returned by our decision support system.

Further, we evaluate the performance of Algorithm 1 in identifying the optimal arm ε^* using our human subject study data. To this end, we compare its simple regret against the simple regret achieved by the uniform discretization algorithm (Slivkins et al., 2019) under several values of the exploration budget n. For completeness, we include the uniform discretization algorithm under Algorithm 2. Here, we implement Algorithm 2 using 100 discretization levels, *i.e.*, 100 equally sized intervals covering the entire range of ε values and distribute the exploration budget uniformly across those. Figure 5 summarizes the results, which show that the simple regret of our algorithm converges to zero as the exploration budget n increases, as expected. Moreover, they also show that, for the same exploration budget n, our algorithm achieves lower simple regret compared to the uniform discretization algorithm, across the vast majority of n values. Figure 6 complements these results by showing the active intervals in the first and last two iterations for one execution of Algorithm 1. The results demonstrate

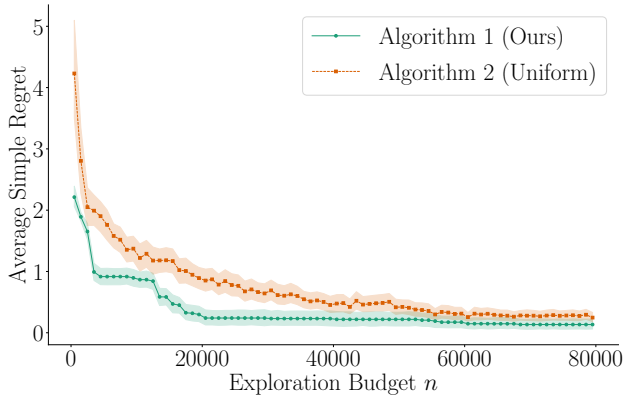


Figure 5: **Simple regret achieved by our Lipschitz best-arm identification algorithm against a uniform discretization baseline.** Each point in the green (orange) line shows the empirical average simple regret achieved over 100 executions of Algorithm 1 (2), respectively, given the respective exploration budget n. Here, we repeat each experiment 100 times using a different random seed for the payoffs obtained from arm pulls and shaded areas represent 95% confidence intervals. In Algorithm 1, we set $\beta = 2$ and $L = 150$.

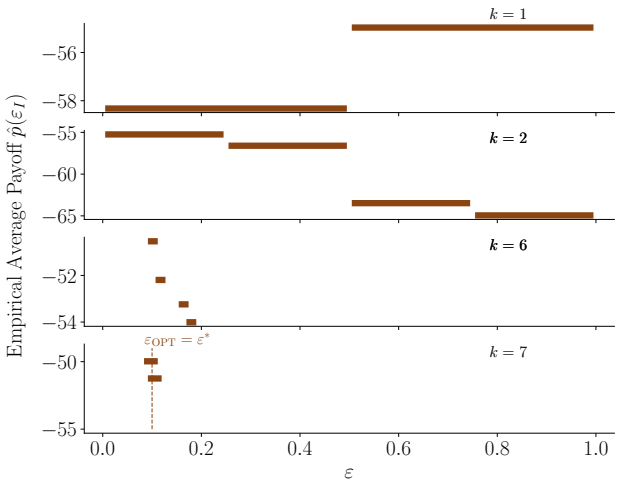


Figure 6: **Evolution of active intervals throughout the execution of Algorithm 1.** From top to bottom, the different plots correspond to the first two and last two iterations of one execution of Algorithm 1. The length of each brown bar on the x-axis corresponds to a range of ε values that form an active interval and its value on the y-axis corresponds to the empirical average payoff the algorithm has obtained by pulling the arm corresponding to that active interval. In the bottom plot, the dashed line indicates the (optimal) value $\varepsilon_{OPT} = \varepsilon^*$, which was successfully identified by the algorithm. Here, we set the parameters of Algorithm 1 to $n = 30{,}000$, $\beta = 2$, and $L = 150$.

that, in practice, Algorithm 1 successfully zooms in the interval containing the optimal value ε^*, and identifies it as the optimal.

5 DISCUSSION AND LIMITATIONS

In this section, we highlight several limitations of our work and discuss avenues for future research as well as its broader impact.

Methodology. Our decision support system utilizes the same parameter value ε for each state of the environment. However, one may obtain improved performance by utilizing a different parameter value $\varepsilon(\mathbf{s})$ per state $\mathbf{s}$. Algorithm 1 requires as input the Lipschitz constant L, a common requirement among Lipschitz bandit algorithms. Overcoming this requirement by developing decision support systems and bandit algorithms that do not require a priori knowledge of the Lipschitz constant L is a challenging yet very interesting avenue for future work. By design, our methodology ensures that a human who takes action supported by our decision support system achieves better or equal average performance than the human or the AI agent used by our system on their own. However, it would be very important to better understand the factors that influence the degree of human-AI complementarity offered by our system. Finally, our decision support system requires actions to be discrete. It would be interesting to lift this requirement and allow for continuous actions.

Algorithm 2 Uniform Discretization

Input: Exploration budget n, discretization levels $\mathcal{D}$.
for $I \in \mathcal{D}$ **do**
  $\varepsilon_I \leftarrow \text{midpoint}(I)$
  $\hat{p}(\varepsilon_I) \leftarrow \textsc{PullArm}(\varepsilon_I, n/|\mathcal{D}|)$ // Pull the midpoint of each interval $n/|\mathcal{D}|$ times
end
$\hat{p}_{\max} \leftarrow \max_{I \in \mathcal{D}} \hat{p}(\varepsilon_I)$
$\varepsilon_{\text{OPT}} \leftarrow \textsc{Midpoint}\left(\text{argmax}_{I \in \mathcal{D}} \hat{p}(\varepsilon_I)\right)$
// Find the interval whose midpoint achieves the highest average payoff
return ε_{OPT}

Evaluation. Our large-scale human subject study suggests our decision support system can enable human-AI complementarity in sequential decision making tasks. However, it comprises only one sequential decision making task—a toy wildfire mitigation game—and compares our decision support system only against the two most natural baselines, in which either the human or the AI takes all actions on their own. It would be important to conduct additional human subject studies comprising other (real) sequential decision making tasks and comparing against alternative approaches

to decision support, such as simple top-k action recommendations or confidence-based filtering (where the human takes actions only when the AI is uncertain). However, it is worth highlighting that conducting such studies at scale would entail significant financial costs—the total cost of our human subject study was 5,280 GBP.

Broader Impact. Our work has focused on maximizing the average cumulative reward achieved by a human using a decision support system based on action sets. However, whenever the decision support system is used in high-stakes domains, it would be important to extend our methodology to account for safety and fairness considerations.

6 CONCLUSIONS

In this paper, we have developed a decision support system for sequential decision making tasks that do not require human experts to understand when to cede agency to the system to achieve human-AI complementarity. Along the way, to optimize the design of our system, we have also introduced and analyzed a novel Lipschitz bandit algorithm with sublinear regret guarantees, which may be of independent interest. Further, we have conducted a large-scale human subject study with hundreds of participants to experimentally validate that our system achieves human-AI complementarity, outperforming both humans and AI on their own, in a (non-trivial) sequential decision making task—a wildfire mitigation game.

Acknowledgements

Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreements No. 945719 and 101169607). Straitouri acknowledges support from a Google PhD Fellowship. Tsirtsis acknowledges support from the Alexander von Humboldt Foundation in the framework of the Alexander von Humboldt Professorship (Humboldt Professor of Technology and Regulation awarded to Sandra Wachter) endowed by the Federal Ministry of Education and Research via the Hasso Plattner Institute. We would also like to give credit to the creator Freepik from `https://www.flaticon.com` whose icons we have used to design our experiment.

References

Rajeev Agrawal. The continuum-armed bandit problem. *SIAM journal on control and optimization*, 33(6):1926–1951, 1995.

Rohan Alur, Loren Laine, Darrick Li, Manish Raghavan, Devavrat Shah, and Dennis Shung. Auditing for human expertise. *Advances in Neural Information Processing Systems*, 36:79439–79468, 2023.

Adrian Arnaiz-Rodriguez, Nina Corvelo Benz, Suhas Thejaswi, Nuria Oliver, and Manuel Gomez-Rodriguez. Towards human-ai complementarity in matching tasks. *arXiv preprint arXiv:2508.13285*, 2025.

Vahid Balazadeh, Abir De, Adish Singla, and Manuel Gomez-Rodriguez. Learning to switch among agents in a team via 2-layer markov decision processes. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL `https://openreview.net/forum?id=NT9zgedd3I`.

Gagan Bansal, Besmira Nushi, Ece Kamar, Eric Horvitz, and Daniel S Weld. Is the most accurate ai the best teammate? optimizing ai for teamwork. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(13): 11405–11414, 2021a.

Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI conference on human factors in computing systems*, pages 1–16, 2021b.

Hamsa Bastani, Osbert Bastani, and Wichinpong Park Sinchaisri. Improving human sequential decision making with reinforcement learning. *Management Science*, 2025.

Viktor Bengs and Eyke Hüllermeier. Preselection bandits. In *International Conference on Machine Learning*, pages 778–787. PMLR, 2020.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

C Brodley, A Kak, C Shyu, J Dy, L Broderick, and Alex M Aisen. Content-based retrieval from medical image databases: A synergy of human interaction, machine learning and computer vision. In *AAAI/IAAI*, pages 760–767, 1999.

Sébastien Bubeck, Gilles Stoltz, Csaba Szepesvári, and Rémi Munos. Online optimization in x-armed bandits. *Advances in Neural Information Processing Systems*, 21, 2008.

Prithwiraj Choudhury, Evan Starr, and Rajshree Agarwal. Machine learning and human capital complementarities: Experimental evidence on bias mitigation. *Strategic Management Journal*, 41(8):1381–1411, 2020.

Nina Corvelo Benz and Manuel Gomez-Rodriguez. Human-aligned calibration for ai-assisted decision making. *Advances in Neural Information Processing Systems*, 36: 14609–14636, 2023.

Nina Corvelo Benz and Manuel Gomez-Rodriguez. Human-alignment influences the utility of ai-assisted decision making. *Scientific Reports*, 15(1):29154, 2025.

Abir De, Paramita Koley, Niloy Ganguly, and Manuel Gomez-Rodriguez. Regression under human assistance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(03):2611–2620, 2020.

Abir De, Nastaran Okati, Ali Zarezade, and Manuel Gomez Rodriguez. Classification under human assistance. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):5905–5913, 2021.

Yasong Feng, Weijian Luo, Yimin Huang, and Tianyu Wang. A lipschitz bandits approach for continuous hyperparameter optimization. *arXiv preprint arXiv:2302.01539*, 2023.

Julien Grand-Clément and Jean Pauphilet. The best decisions are not the best advice: Making adherence-aware recommendations. *Management Science*, 2024.

Eric Horvitz and Tim Paek. Complementary computing: policies for transferring callers from dialog systems to human receptionists. *User Modeling and User-Adapted Interaction*, 17:159–182, 2007.

Shengyi Huang and Santiago Ontañón. A closer look at invalid action masking in policy gradient algorithms. In *FLAIRS*. Florida Online Journals, 2022.

Nehemiah Jordan. Allocation of functions between man and machines in automated systems. *Journal of applied psychology*, 47(3):161, 1963.

Ece Kamar, Severin Hacker, and Eric Horvitz. Combining human and machine intelligence in large-scale crowdsourcing. In *Aamas*, volume 12, pages 467–474, 2012.

Gavin Kerrigan, Padhraic Smyth, and Mark Steyvers. Combining human predictions with model probabilities via confusion matrices and calibration. *Advances in Neural Information Processing Systems*, 34:4421–4434, 2021.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 681–690, 2008.

Vivian Lai, Chacha Chen, Q Vera Liao, Alison Smith-Renner, and Chenhao Tan. Towards a science of human-ai decision making: a survey of empirical studies. *arXiv preprint arXiv:2112.11471*, 2021.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. *Advances in Neural Information Processing Systems*, 35:31199–31212, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Hussein Mozannar and David Sontag. Consistent estimators for learning to defer to an expert. In *International conference on machine learning*, pages 7076–7087. PMLR, 2020.

Arthur Müller and Matthia Sabatelli. Safe and psychologically pleasant traffic signal control with reinforcement learning using action masking. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 951–958. IEEE, 2022.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

Nastaran Okati, Abir De, and Manuel Rodriguez. Differentiable learning under triage. *Advances in Neural Information Processing Systems*, 34:9140–9151, 2021.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Si-Ze Qian, Yuhong Xie, Zipeng Pan, Yuan Zhang, and Tao Lin. Dam: Deep reinforcement learning based preload algorithm with action masking for short video streaming. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 7030–7034, 2022.

Maithra Raghu, Katy Blumer, Greg Corrado, Jon Kleinberg, Ziad Obermeyer, and Sendhil Mullainathan. The algorithmic automation problem: Prediction, triage, and human effort. *arXiv preprint arXiv:1903.12220*, 2019.

Aleksandrs Slivkins et al. Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning*, 12(1-2):1–286, 2019.

Mark Steyvers, Heliodoro Tejeda, Gavin Kerrigan, and Padhraic Smyth. Bayesian modeling of human–ai complementarity. *Proceedings of the National Academy of Sciences*, 119(11):e2111547119, 2022.

Eleni Straitouri and Manuel Gomez-Rodriguez. Designing decision support systems using counterfactual prediction sets. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 46722–46744. PMLR, 21–27 Jul 2024.

Eleni Straitouri, Adish Singla, Vahid Balazadeh Meresht, and Manuel Gomez-Rodriguez. Reinforcement learning under algorithmic triage. *arXiv preprint arXiv:2109.11328*, 2021.

Eleni Straitouri, Lequn Wang, Nastaran Okati, and Manuel Gomez-Rodriguez. Improving expert predictions with conformal prediction. In *International Conference on Machine Learning*, pages 32633–32653. PMLR, 2023.

Gail M Sullivan and Richard Feinn. Using effect size—or why the p value is not enough. *Journal of graduate medical education*, 4(3):279–282, 2012.

Harini Suresh, Natalie Lao, and Ilaria Liccardi. Misplaced trust: Measuring the interference of machine learning in human decision-making. In *Proceedings of the ACM Conference on Web Science*, pages 315–324. ACM, 2020.

Giovanni De Toni, Nastaran Okati, Suhas Thejaswi, Eleni Straitouri, and Manuel Gomez-Rodriguez. Towards human-AI complementarity with prediction sets. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=xtK3gZjQDC`.

Stratis Tsirtsis, Abir De, and Manuel Gomez-Rodriguez. Counterfactual explanations in sequential decision making under uncertainty. *Advances in Neural Information Processing Systems*, 34:30127–30139, 2021.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *Proceedings of the AAAI conference on artificial intelligence*, 30 (1), 2016.

Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

Ziyi Wang, Xinran Li, Luoyang Sun, Haifeng Zhang, Hualin Liu, and Jun Wang. Learning state-specific action masks for reinforcement learning. *Algorithms*, 17(2):60, 2024.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. Mastering complex control in moba games with deep reinforcement learning. *Proceedings of the AAAI conference on artificial intelligence*, 34(04): 6672–6679, 2020.

Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. Understanding the effect of accuracy on trust in machine learning models. In *Proceedings of the CHI conference on human factors in computing systems*, pages 1–12, 2019.

Qianqian Zhang, Yu Kang, Yun-Bo Zhao, Pengfei Li, and Shiyi You. Traded control of human–machine systems for sequential decision-making based on reinforcement learning. *IEEE Transactions on Artificial Intelligence*, 3 (4):553–566, 2021.

Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 295–305. ACM, 2020.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] Section 2, 3, and Appendix B.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] Section 3.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] Submitted as supplementary material.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes] Section 2, 3.
   (b) Complete proofs of all theoretical results. [Yes] Appendix A.
   (c) Clear explanations of any assumptions. [Yes] Section 2, 3.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] Submitted as supplementary material.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] Section 4, Appendix B.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] Section 4.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] Section 4.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. [Yes] Appendix B.
   (b) The license information of the assets, if applicable. [Yes] Appendix B.
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes] Our code and human study data are submitted as supplementary material.
   (d) Information about consent from data providers/curators. [Yes] Appendix B.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable] As we report in Section 4, we do not record any personally identifiable information.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Yes] Appendix B.
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Yes] We include an anonymized reference to our IRB approval not to violate anonymity. We are happy to provide the IRB approval to the area chair if requested.
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes] Section 4, 5.

A PROOFS

A.1 Proof of Proposition 1

Given a ranking $a_{(1)}, a_{(2)}, \ldots, a_{(m)}$ of the m actions based on the (scaled) valuations $\tilde{q}(\mathbf{s}, a)$ of the AI agent, let Δ_i be the difference between the highest and the i-th highest (scaled) valuation, *i.e.*, $\Delta_i = \tilde{q}(\mathbf{s}, a_{(1)}) - \tilde{q}(\mathbf{s}, a_{(i)})$ for $i = 1, 2, \ldots, m$. For ease of exposition, we will slightly abuse the notation and consider that there is a dummy action $a_{(m+1)}$ with valuation $\tilde{q}(\mathbf{s}, a_{(m+1)}) = -\infty$ and, consequently, $\Delta_{m+1} = +\infty$. Then, we can write

$$\begin{aligned}
\mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] &\overset{(*)}{=} \mathbb{P}\left[\left(\tilde{q}(\mathbf{s}, a_{(i)}) + W \geq \tilde{q}(\mathbf{s}, a_{(1)}) - \varepsilon\right) \wedge \left(\tilde{q}(\mathbf{s}, a_{(i+1)}) + W < \tilde{q}(\mathbf{s}, a_{(1)}) - \varepsilon\right)\right] \\
&= \mathbb{P}\left[\tilde{q}(\mathbf{s}, a_{(1)}) - \tilde{q}(\mathbf{s}, a_{(i)}) - \varepsilon \leq W < \tilde{q}(\mathbf{s}, a_{(1)}) - \tilde{q}(\mathbf{s}, a_{(i+1)}) - \varepsilon\right] \\
&= \mathbb{P}\left[\Delta_i - \varepsilon \leq W < \Delta_{i+1} - \varepsilon\right] \\
&= \int_{\Delta_i - \varepsilon}^{\Delta_{i+1} - \varepsilon} g_W(w) dw
\end{aligned}$$

where $(*)$ follows by the definition of the decision support policy in Eq. 3 and g_W is the probability density function of W. Now, without loss of generality, assume that $\varepsilon < \varepsilon'$. Then, it follows that

$$\begin{aligned}
\left|\mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] - \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}]\right| &= \left|\int_{\Delta_i - \varepsilon}^{\Delta_{i+1} - \varepsilon} g_W(w) dw - \int_{\Delta_i - \varepsilon'}^{\Delta_{i+1} - \varepsilon'} g_W(w) dw\right| \\
&\overset{(*)}{=} \left|\int_{\Delta_i}^{\Delta_{i+1}} g_W(x - \varepsilon) dx - \int_{\Delta_i}^{\Delta_{i+1}} g_W(x - \varepsilon') dx\right| \\
&\overset{(**)}{\leq} \int_{\Delta_i}^{\Delta_{i+1}} |g_W(x - \varepsilon) - g_W(x - \varepsilon')| dx,
\end{aligned}$$

where in $(*)$ we made the change of variable $x = w + \varepsilon$ (ε') in the first (second) integral and in $(**)$ we used the triangle inequality. Summing over all $i = 1, 2, \ldots, m$, we get that

$$\begin{aligned}
\sum_{i=1}^{m} \left|\mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] - \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}]\right| &\leq \sum_{i=1}^{m} \int_{\Delta_i}^{\Delta_{i+1}} |g_W(x - \varepsilon) - g_W(x - \varepsilon')| dx \\
&= \int_0^{+\infty} |g_W(x - \varepsilon) - g_W(x - \varepsilon')| dx \overset{(*)}{=} \int_{-\varepsilon'}^{+\infty} |g_W(y + (\varepsilon' - \varepsilon)) - g_W(y)| dy \\
&= \int_{-\varepsilon'}^{0} |g_W(y + (\varepsilon' - \varepsilon)) - g_W(y)| dy + \int_0^{+\infty} |g_W(y + (\varepsilon' - \varepsilon)) - g_W(y)| dy \\
&\overset{(**)}{\leq} \int_{-\varepsilon}^{\varepsilon' - \varepsilon} |g_W(y')| dy' + \int_{-\varepsilon'}^{0} |g_W(y)| dy + \int_0^{+\infty} |g_W(y + (\varepsilon' - \varepsilon)) - g_W(y)| dy \\
&\overset{(***)}{=} \int_0^{\varepsilon' - \varepsilon} |g_W(y')| dy' + \int_0^{+\infty} |g_W(y + (\varepsilon' - \varepsilon)) - g_W(y)| dy, \quad (7)
\end{aligned}$$

where in $(*)$ we made the change of variable $y = x - \varepsilon'$, in $(**)$ we used the triangle inequality and made the change of variable $y' = y + (\varepsilon' - \varepsilon)$, and in $(***)$ we used the fact that $g_W(y) = 0$ for all $y < 0$ since W follows a half-normal distribution. Since the probability density function of a half-normal distribution is given by $g_W(y) = \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \cdot e^{-y^2/2\sigma^2}$ for $y \geq 0$, it is easy to verify that $g_W(y)$ is upper bounded by $\frac{\sqrt{2}}{\sigma\sqrt{\pi}}$ and, therefore, the first term on the right-hand side of the above inequality is upper bounded by $\frac{\sqrt{2}}{\sigma\sqrt{\pi}} \cdot (\varepsilon' - \varepsilon)$.

To bound the second term, we use the fundamental theorem of calculus, which gives that, for any $y \geq 0$,

$$|g_W(y + (\varepsilon' - \varepsilon)) - g_W(y)| = \left|\int_y^{y + (\varepsilon' - \varepsilon)} g'_W(x) dx\right| \overset{(*)}{\leq} \int_0^{\varepsilon' - \varepsilon} |g'_W(\theta + y)| \, d\theta,$$

where in $(*)$ we made the change of variable $\theta = x - y$ and used the triangle inequality. Hence, we get

$$\int_0^{+\infty} |g_W(y + (\varepsilon' - \varepsilon)) - g_W(y)| dy \leq \int_0^{+\infty} \int_0^{\varepsilon' - \varepsilon} |g'_W(\theta + y)| d\theta dy = \int_0^{\varepsilon' - \varepsilon} \int_0^{+\infty} |g'_W(\theta + y)| dy d\theta$$
$$\overset{(*)}{\leq} \int_0^{\varepsilon' - \varepsilon} \int_0^{+\infty} |g'_W(y)| dy d\theta = (\varepsilon' - \varepsilon) \cdot \int_0^{+\infty} |g'_W(y)| dy = (\varepsilon' - \varepsilon) \cdot \int_0^{+\infty} \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \frac{y}{\sigma^2} \cdot e^{-y^2/2\sigma^2} dy$$
$$\overset{(**)}{=} (\varepsilon' - \varepsilon) \cdot \frac{\sqrt{2}}{\sigma\sqrt{\pi}} \cdot \int_0^{+\infty} e^{-u} du = (\varepsilon' - \varepsilon) \cdot \frac{\sqrt{2}}{\sigma\sqrt{\pi}},$$

where $(*)$ follows from the fact that $\theta, y, |g'_W| \geq 0$ and in $(**)$ we made the change of variable $u = y^2/2\sigma^2$. Having bounded both terms on the right-hand side of Eq. 7 with $\frac{\sqrt{2}}{\sigma\sqrt{\pi}} \cdot (\varepsilon' - \varepsilon)$, we can conclude that the proposition holds with $L_c = \frac{2\sqrt{2}}{\sigma\sqrt{\pi}}$.

A.2 Proof of Proposition 2

To prove the proposition, it suffices to show that, for any $\mathbf{z} \in \mathcal{Z}$ and $\varepsilon, \varepsilon' \in [0, 1]$, it holds that

$$|v(\mathbf{z}; \varepsilon) - v(\mathbf{z}; \varepsilon')| \leq L \cdot |\varepsilon - \varepsilon'| . \tag{8}$$

Once that is established, using Jensen's inequality, we are able to get

$$|\mathbb{E}_{\boldsymbol{Z}_1}[v(\boldsymbol{Z}_1; \varepsilon)] - \mathbb{E}_{\boldsymbol{Z}_1}[v(\boldsymbol{Z}_1; \varepsilon')]| \leq \mathbb{E}_{\boldsymbol{Z}_1}[|v(\boldsymbol{Z}_1; \varepsilon) - v(\boldsymbol{Z}_1; \varepsilon')|] = \int_{\mathcal{Z}} |v(\mathbf{z}; \varepsilon) - v(\mathbf{z}; \varepsilon')| \cdot f_{Z_1}(\mathbf{z}) d\mathbf{z}$$
$$\leq L \cdot |\varepsilon - \varepsilon'| \int_{\mathcal{Z}} f_{Z_1}(\mathbf{z}) d\mathbf{z} = L \cdot |\varepsilon - \varepsilon'|,$$

where $f_{\boldsymbol{Z}_1}$ is the probability density function of the initial state $\boldsymbol{Z}_1$.

The main idea of the proof is to first show a variant of Eq. 8 for a finite horizon T, and then extend it to the infinite-horizon case by taking the limit $T \to \infty$. For the finite-horizon case, our goal is to show that, for any horizon $T \geq 1$, there exists a constant $L_T > 0$ such that for all $\mathbf{z} \in \mathcal{Z}$ and $\varepsilon, \varepsilon' \in [0, 1]$ it holds that

$$|v_T(\mathbf{z}; \varepsilon) - v_T(\mathbf{z}; \varepsilon')| \leq L_T \cdot |\varepsilon - \varepsilon'|, \tag{9}$$

where $v_T(\mathbf{z}; \varepsilon) = \mathbb{E}_{\boldsymbol{W}, \boldsymbol{B}, \boldsymbol{U}} \left[\sum_{t=0}^{T-1} \gamma^t \cdot R_{t+1} \mid \boldsymbol{Z}_1 = \mathbf{z} \right]$.

We will prove that Eq. 9 holds for any finite horizon T by induction. We start with the base case $T = 1$, where the quantity $v_1(\mathbf{z}; \varepsilon)$ corresponds to the immediate reward obtained by the human, depending on the state $\mathbf{z}$ and its representation $\mathbf{s} = f_S(\mathbf{z})$, the action set $\mathcal{C}$ provided by the decision support system, and the action A they choose from that action set. Similarly to the proof of Proposition 1, we will use the fact that the decision support policy $\pi_{\mathcal{C}}$ can assign non-zero probability only to action sets $\mathcal{C}$ that are of the form $\mathcal{C}_{(i)} = \{a_{(1)}, a_{(2)}, \ldots, a_{(i)}\}$, where $a_{(1)}, a_{(2)}, \ldots, a_{(m)}$ is the ranking of the m actions based on the valuations $q(\mathbf{s}, a)$ of the AI agent. Then, we can write

$$v_1(\mathbf{z}; \varepsilon) = \sum_{i=1}^{m} \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] \sum_{a \in \mathcal{A}} \mathbb{P}[A = a \mid \boldsymbol{Z} = \mathbf{z}, \mathcal{C} = \mathcal{C}_{(i)}] \cdot f_R(\mathbf{z}, a, \varnothing),$$

where we have used $f_R(\mathbf{z}, a, \varnothing)$ to denote the reward obtained by the human when they take an action a in a state $\mathbf{z}$ right before the end of the episode. Following from that, we get

$$|v_1(\mathbf{z}; \varepsilon) - v_1(\mathbf{z}; \varepsilon')| \leq \sum_{i=1}^{m} \left| \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] - \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \right|$$
$$\cdot \sum_{a \in \mathcal{A}} \mathbb{P}[A = a \mid \boldsymbol{Z} = \mathbf{z}, \mathcal{C} = \mathcal{C}_{(i)}] \cdot |f_R(\mathbf{z}, a, \varnothing)|$$
$$\overset{(*)}{\leq} \sum_{i=1}^{m} \left| \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] - \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \right| \cdot r_{max}$$
$$\overset{(**)}{\leq} r_{max} \cdot L_c \cdot |\varepsilon - \varepsilon'|,$$

where $(*)$ follows from the fact that the rewards are bounded by r_{max}, and $(**)$ follows from Proposition 1. Therefore, the base case holds with $L_1 = r_{max} \cdot L_c$.

Now, we will prove the inductive step. Assume that Eq. 9 holds for $T = n-1$ for some natural number $n \geq 2$, that is, there exists $L_{n-1} > 0$ such that, for all $\mathbf{z} \in \mathcal{Z}$ and $\varepsilon, \varepsilon' \in [0,1]$, it holds that

$$|v_{n-1}(\mathbf{z};\varepsilon) - v_{n-1}(\mathbf{z};\varepsilon')| \leq L_{n-1} \cdot |\varepsilon - \varepsilon'|.$$

We will show that it also holds for $T = n$. First, we can write

$$\begin{aligned}
v_n(\mathbf{z};\varepsilon) &= \sum_{i=1}^{m} \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] \cdot (f_R(\mathbf{z}, a, \mathbf{z}') + \gamma \cdot v_{n-1}(\mathbf{z}';\varepsilon))\, d\mathbf{z}' \\
&= \underbrace{\sum_{i=1}^{m} \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] f_R(\mathbf{z}, a, \mathbf{z}') d\mathbf{z}'}_{\text{Immediate reward term } (\dagger\,;\,\varepsilon)} \\
&\quad + \underbrace{\gamma \sum_{i=1}^{m} \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] \cdot v_{n-1}(\mathbf{z}';\varepsilon) d\mathbf{z}'}_{\text{Future discounted cumulative reward term } (\ddagger\,;\,\varepsilon)},
\end{aligned}$$

where, for brevity, we have denoted as $\mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}]$ the probability that the human takes action a in state $\mathbf{z}$ when they are provided with action set $\mathcal{C}_{(i)}$ and as $\mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a]$ the transition probability from state $\mathbf{z}$ to state $\mathbf{z}'$ when the human takes action a. Further, for ease of exposition, we will bound separately the absolute differences of the two terms $(\dagger)$ and $(\ddagger)$ resulting from $\varepsilon, \varepsilon'$. Starting with the immediate reward term, we have

$$\begin{aligned}
|(\dagger\,;\,\varepsilon) - (\dagger\,;\,\varepsilon')| &\overset{(*)}{\leq} \sum_{i=1}^{m} \left| \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] - \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \right| \\
&\qquad \cdot \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] \cdot |f_R(\mathbf{z}, a, \mathbf{z}')| d\mathbf{z}' \\
&\overset{(**)}{\leq} \sum_{i=1}^{m} \left| \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] - \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \right| r_{max} \overset{(***)}{\leq} r_{max} \cdot L_c \cdot |\varepsilon - \varepsilon'|,
\end{aligned}$$

where $(*)$ follows from the triangle inequality, $(**)$ follows from the fact that the rewards are bounded by r_{max}, and $(***)$ follows from Proposition 1.

Next, we will bound the difference of the future discounted cumulative reward terms. We can write it as

$$\begin{aligned}
|(\ddagger\,;\,\varepsilon) - (\ddagger\,;\,\varepsilon')| = \Bigg| &\gamma \sum_{i=1}^{m} \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] \cdot v_{n-1}(\mathbf{z}';\varepsilon) d\mathbf{z}' \\
&- \gamma \sum_{i=1}^{m} \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] \cdot v_{n-1}(\mathbf{z}';\varepsilon') d\mathbf{z}' \\
&\pm \gamma \sum_{i=1}^{m} \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] \cdot v_{n-1}(\mathbf{z}';\varepsilon) d\mathbf{z}' \Bigg| \\
\overset{(*)}{\leq}\ &\gamma \sum_{i=1}^{m} \left| \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon) = \mathcal{C}_{(i)}] - \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \right| \\
&\qquad \cdot \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] \cdot |v_{n-1}(\mathbf{z}';\varepsilon)| d\mathbf{z}' \\
&+ \gamma \sum_{i=1}^{m} \mathbb{P}[\pi_{\mathcal{C}}(\mathbf{s}, W; \varepsilon') = \mathcal{C}_{(i)}] \\
&\qquad \cdot \sum_{a \in \mathcal{A}} \mathbb{P}[a \,|\, \mathbf{z}, \mathcal{C}_{(i)}] \int_{\mathcal{Z}} \mathbb{P}[\mathbf{z}' \,|\, \mathbf{z}, a] \cdot |v_{n-1}(\mathbf{z}';\varepsilon) - v_{n-1}(\mathbf{z}';\varepsilon')| d\mathbf{z}'
\end{aligned}$$

$$\overset{(**)}{\leq} \gamma \cdot L_c \cdot |\varepsilon - \varepsilon'| \cdot \frac{r_{max}}{1-\gamma} + \gamma \cdot L_{n-1} \cdot |\varepsilon - \varepsilon'|,$$

where we have used red color for ε and blue color for ε' for ease of exposition. Here, $(*)$ follows from the triangle inequality and in $(**)$ we used Proposition 1, the induction hypothesis, and the fact that

$$|v_{n-1}(\mathbf{z}'; \varepsilon)| \leq \sum_{t=0}^{n-2} \gamma^t \cdot r_{max} \leq \sum_{t=0}^{\infty} \gamma^t \cdot r_{max} = \frac{r_{max}}{1-\gamma}.$$

Combining the bounds for the two terms, we can conclude that

$$\begin{aligned} |v_n(\mathbf{z}; \varepsilon) - v_n(\mathbf{z}; \varepsilon')| &\leq \left(r_{max} \cdot L_c + \gamma \cdot L_c \cdot \frac{r_{max}}{1-\gamma} + \gamma \cdot L_{n-1} \right) \cdot |\varepsilon - \varepsilon'| \\ &= \left(L_c \cdot \frac{r_{max}}{1-\gamma} + \gamma \cdot L_{n-1} \right) \cdot |\varepsilon - \varepsilon'|, \end{aligned}$$

which yields the closed form expression

$$L_n = L_c \cdot r_{max} \cdot \left[\gamma^{n-1} + \frac{1-\gamma^{n-1}}{1-\gamma} \cdot \frac{1}{1-\gamma} \right] = L_c \cdot r_{max} \cdot \frac{\gamma^n(\gamma-2)+1}{(1-\gamma)^2}.$$

As a result, we have shown that Eq. 9 holds for any finite horizon T. Now, we will extend the result to the infinite-horizon case. First, it is easy to see that the limit $\lim_{T\to\infty} v_T(\mathbf{z}; \varepsilon)$ is finite for all $\mathbf{z} \in \mathcal{Z}$ and $\varepsilon \in [0, 1]$ since $|v_T(\mathbf{z}; \varepsilon)| \leq \sum_{t=0}^{T-1} \gamma^t \cdot r_{max} \leq \frac{r_{max}}{1-\gamma}$ for all $T \geq 1$. Therefore, we can write

$$\begin{aligned} |v(\mathbf{z}; \varepsilon) - v(\mathbf{z}; \varepsilon')| = \left| \lim_{T\to\infty} v_T(\mathbf{z}; \varepsilon) - \lim_{T\to\infty} v_T(\mathbf{z}; \varepsilon') \right| &= \left| \lim_{T\to\infty} (v_T(\mathbf{z}; \varepsilon) - v_T(\mathbf{z}; \varepsilon')) \right| \\ &\leq \lim_{T\to\infty} |v_T(\mathbf{z}; \varepsilon) - v_T(\mathbf{z}; \varepsilon')| \overset{(*)}{\leq} \lim_{n\to\infty} L_n \cdot |\varepsilon - \varepsilon'|, \end{aligned}$$

where in $(*)$ we used the result for the finite horizon case. Finally, it is easy to verify that $\lim_{n\to\infty} L_n = \frac{L_c \cdot r_{max}}{(1-\gamma)^2}$, which concludes the proof of the proposition for the infinite horizon case with $L = \frac{L_c \cdot r_{max}}{(1-\gamma)^2}$.

A.3 Proof of Proposition 3

For ease of exposition, here we assume that all observed payoffs take values in $[0, 1]$. Note that this is without loss of generality, since the reward at each time step lies in the bounded interval $[-r_{max}, r_{max}]$ and, consequently, the (discounted) cumulative reward is also bounded. Therefore, this assumption has an effect only on constants in the simple regret guarantees of Algorithm 1 that we derive next.

We will prove the proposition by distinguishing between two types of events: a "clean" event that holds with high probability and the complementary "bad" event, similarly to the analysis of other bandit algorithms (Slivkins et al., 2019). Our goal is to first establish a lower bound on the probability of the clean event, and then show that, if the clean event holds, then the algorithm returns an arm ε_{ALG} with a sufficiently small simple regret, *i.e.*, $p(\varepsilon^*) - p(\varepsilon_{ALG})$.

Formally, let E denote the clean event where, for all iterations of the algorithm and all active intervals, the estimates of the average payoffs for all arms are sufficiently accurate, *i.e.*,

$$E = \bigcap_{k=1}^{k_{max}} \bigcap_{I \in \mathcal{I}_k} \left\{ |\hat{p}(\varepsilon_I) - p(\varepsilon_I)| \leq l_k \right\},$$

where k_{max} is the maximum number of iterations of the algorithm, $l_k = 2^{-k}$ is the length of the intervals at iteration k, and ε_I is the midpoint of interval I.

We will start by showing that the number of iterations k_{max} is bounded. Note that, at each iteration k, the algorithm samples $n_k = 2^{k\beta}$ times at each active interval, while the total number of samples available to the

algorithm is n. Therefore, in the worst case where the algorithm eliminates all but one interval at each iteration, the maximum number of iterations k_{max} has to satisfy

$$n \geq \sum_{k=1}^{k_{max}} n_k = \sum_{k=1}^{k_{max}} 2^{k\beta} = \frac{2^\beta(2^{k_{max}\beta}-1)}{2^\beta - 1} \Rightarrow k_{max} := \left\lceil \frac{\log_2(n(2^\beta-1)+1)}{\beta} \right\rceil.$$

Now, we will establish a lower bound on the probability of the clean event E. Since for each interval $I \in \mathcal{I}_k$, the estimate $\hat{p}(\varepsilon_I)$ is obtained by averaging n_k payoff samples, Hoeffding's inequality implies that

$$\mathbb{P}[|\hat{p}(\varepsilon_I) - p(\varepsilon_I)| > l_k] \leq 2\exp(-2n_k l_k^2).$$

Moreover, since the samples drawn to estimate the payoffs at the midpoints of the intervals in each iteration are independent of the algorithm's choices in previous iterations, the events $\{|\hat{p}(\varepsilon_I) - p(\varepsilon_I)| \leq l_k\}$ for all $I \in \mathcal{I}_k$ are independent. Therefore, the probability of the clean event E can be bounded as follows:

$$\begin{aligned}
\mathbb{P}[E] = \mathbb{P}\left[\bigcap_{k=1}^{k_{max}} \bigcap_{I\in\mathcal{I}_k} \{|\hat{p}(\varepsilon_I) - p(\varepsilon_I)| \leq l_k\}\right] &= 1 - \mathbb{P}\left[\bigcup_{k=1}^{k_{max}} \bigcup_{I\in\mathcal{I}_k} \{|\hat{p}(\varepsilon_I) - p(\varepsilon_I)| > l_k\}\right] \\
&\overset{(*)}{\geq} 1 - \sum_{k=1}^{k_{max}} \sum_{I\in\mathcal{I}_k} \mathbb{P}[|\hat{p}(\varepsilon_I) - p(\varepsilon_I)| > l_k] \geq 1 - \sum_{k=1}^{k_{max}} \sum_{I\in\mathcal{I}_k} 2\exp(-2n_k l_k^2) \\
&= 1 - 2\sum_{k=1}^{k_{max}} |\mathcal{I}_k| \exp(-2^{k(\beta-2)-1}) \overset{(**)}{\geq} 1 - 2\sum_{k=1}^{k_{max}} 2^k \exp(-2^{k(\beta-2)-1}),
\end{aligned}$$

where $(*)$ follows from the union bound and $(**)$ follows from the fact that $[0,1]$ can be covered by at most 2^k intervals of length $l_k = 2^{-k}$.

Next, we will show that, if the clean event E holds, then the algorithm returns an arm with a sufficiently small simple regret. We start by showing that the optimal arm ε^* is never eliminated during the execution of the algorithm. For the sake of contradiction, assume that, at some iteration k, the interval I_k^* containing the optimal arm ε^* is eliminated. By the definition of the elimination rule, this would mean that there exists an interval $I \in \mathcal{I}_k$ such that

$$\hat{p}(\varepsilon_I) - \hat{p}(\varepsilon_{I_k^*}) > (2 + L/2)l_k.$$

However, since we are considering the case where the clean event E holds, we have that

$$\begin{aligned}
\hat{p}(\varepsilon_I) - \hat{p}(\varepsilon_{I_k^*}) &= \hat{p}(\varepsilon_I) - p(\varepsilon_I) + p(\varepsilon_I) - p(\varepsilon^*) + p(\varepsilon^*) - p(\varepsilon_{I_k^*}) + p(\varepsilon_{I_k^*}) - \hat{p}(\varepsilon_{I_k^*}) \\
&\leq p(\varepsilon_I) - p(\varepsilon^*) + |\hat{p}(\varepsilon_I) - p(\varepsilon_I)| + |p(\varepsilon^*) - p(\varepsilon_{I_k^*})| + |\hat{p}(\varepsilon_{I_k^*}) - p(\varepsilon_{I_k^*})| \\
&\overset{(*)}{\leq} p(\varepsilon_I) - p(\varepsilon^*) + l_k + L \cdot \frac{l_k}{2} + l_k \\
&\overset{(**)}{\leq} (2 + L/2) \cdot l_k,
\end{aligned}$$

where $(*)$ follows from the definition of the clean event and the Lipschitz continuity of p, and $(**)$ follows from the fact that ε^* is optimal. This leads to a contradiction and, hence, intervals containing the optimal arm ε^* are never eliminated during the execution of the algorithm, as long as the clean event E holds.

We will now establish an upper bound on the gap between the payoff of the optimal arm ε^* and the payoffs of the arms in the non-eliminated intervals at each iteration k. Consider a (non-eliminated) interval $I \in \mathcal{I}_k$ and an arm $\varepsilon \in I$. Since I was not eliminated at iteration k, it holds that

$$\hat{p}(\varepsilon_{I_k^*}) - \hat{p}(\varepsilon_I) \leq (2 + L/2)l_k$$

and, since the clean event E holds, we have

$$\hat{p}(\varepsilon_{I_k^*}) \geq p(\varepsilon_{I_k^*}) - l_k, \text{ and } \hat{p}(\varepsilon_I) \leq p(\varepsilon_I) + l_k.$$

Combining the above inequalities, we obtain

$$p(\varepsilon_{I_k^*}) - p(\varepsilon_I) \leq (2 + L/2)l_k + 2l_k = (4 + L/2)l_k,$$

and, using the Lipschitz continuity of p, for any arm $\varepsilon \in I$, we have

$$\begin{aligned} p(\varepsilon^*) - p(\varepsilon) &\leq p(\varepsilon^*) - p(\varepsilon_{I_k^*}) + p(\varepsilon_{I_k^*}) - p(\varepsilon_I) + p(\varepsilon_I) - p(\varepsilon) \\ &\leq L \cdot \frac{l_k}{2} + (4 + L/2)l_k + L \cdot \frac{l_k}{2} \\ &= (4 + 3L/2)l_k. \end{aligned} \tag{10}$$

The above inequality implies that the union of all intervals $I \in \mathcal{I}_k$ is a subset of the set of arms with a payoff within $(4 + 3L/2)l_k$ of the optimal payoff. We will now use the definition of the zooming dimension to bound the number of active intervals $|\mathcal{I}_k|$ at iteration k. Consider a radius $\rho = (4 + 3L/2)l_k$. Then, we know that there exist constants λ and d such that, the set of arms with a payoff within ρ of the optimal payoff can be covered by $N = \lambda \cdot \rho^{-d}$ sets of diameter at most ρ. Since the intervals in $\mathcal{I}_k$ are disjoint and have length $l_k = \frac{\rho}{4+3L/2}$, it has to hold that

$$|\mathcal{I}_k| \cdot l_k \leq N \cdot \rho \Rightarrow |\mathcal{I}_k| \leq (4 + 3L/2) \cdot N \Rightarrow |\mathcal{I}_k| \leq (4 + 3L/2)^{(1-d)} \cdot \lambda \cdot l_k^{-d}.$$

Let K denote the last iteration of the algorithm, that is, the iteration after which the algorithm exhausted its budget n. Since the algorithm terminated right after iteration K, it has to hold that

$$n < \sum_{k=1}^{K+1} |\mathcal{I}_k| n_k \leq \sum_{k=1}^{K+1} (4 + 3L/2)^{(1-d)} \cdot \lambda \cdot l_k^{-d} \cdot n_k = (4 + 3L/2)^{(1-d)} \cdot \lambda \cdot \sum_{k=1}^{K+1} 2^{k(d+\beta)}.$$

Then, we can bound the above sum as follows:

$$\sum_{k=1}^{K+1} 2^{k(d+\beta)} = 2^{d+\beta} \cdot \frac{2^{(K+1)(d+\beta)} - 1}{2^{d+\beta} - 1} \leq 2^{d+\beta} \cdot \frac{2^{(K+1)(d+\beta)}}{2^{d+\beta} - 1} = 2^{d+\beta} \cdot \frac{2^{d+\beta} \cdot l_K^{-(d+\beta)}}{2^{d+\beta} - 1},$$

which, after rearranging the terms, yields

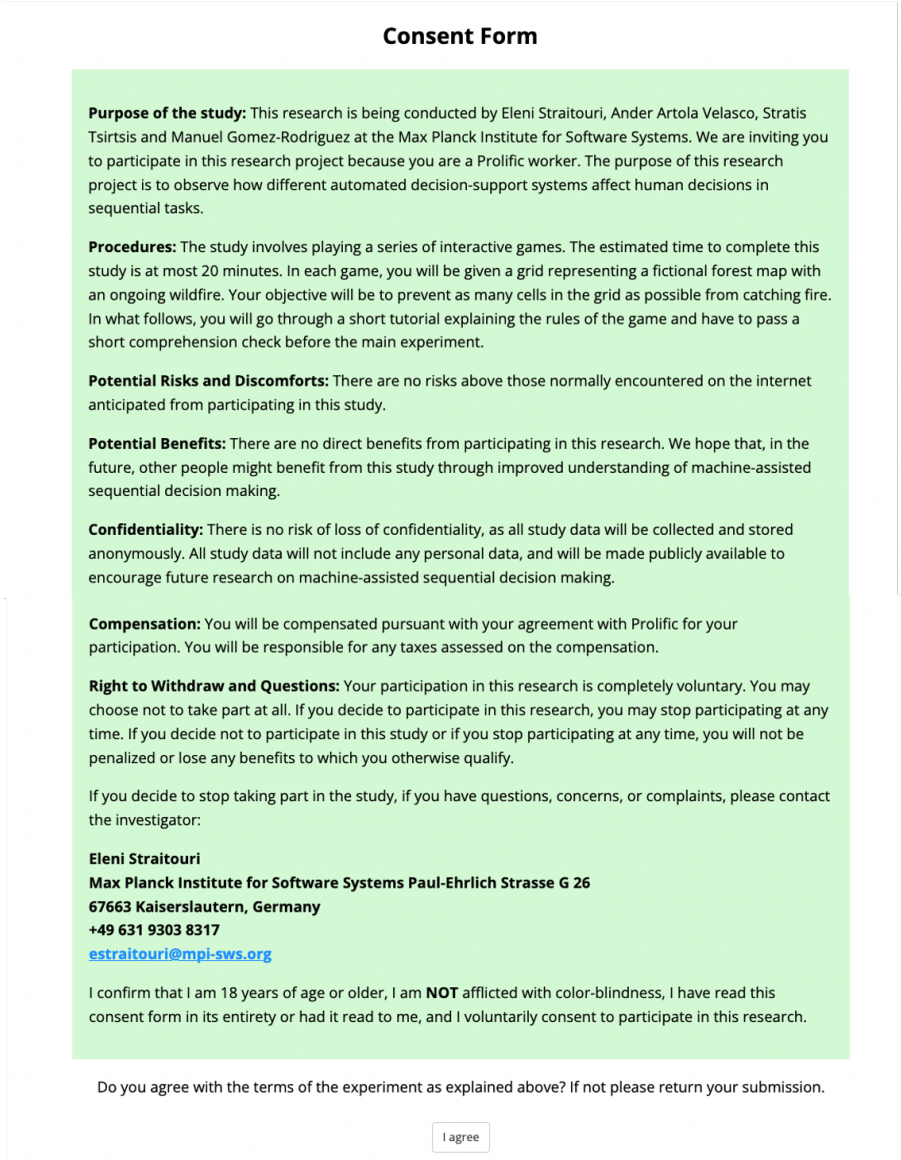
$$l_K \leq n^{\frac{-1}{d+\beta}} \cdot \left((4 + 3L/2)^{(1-d)} \cdot \lambda \cdot \frac{4^{d+\beta}}{2^{d+\beta} - 1} \right)^{\frac{1}{d+\beta}}.$$

Combining the above inequality with Eq. 10 and observing that the arm ε_{ALG} returned by our algorithm comes from an interval not eliminated at iteration K, we get that

$$p(\varepsilon^*) - p(\varepsilon_{ALG}) \leq n^{\frac{-1}{d+\beta}} \cdot (4 + 3L/2)^{\frac{1+\beta}{d+\beta}} \cdot \left(\frac{\lambda \cdot 4^{d+\beta}}{2^{d+\beta} - 1} \right)^{\frac{1}{d+\beta}} = \mathcal{O}\left(n^{\frac{-1}{d+\beta}} \right). \tag{11}$$

This concludes the proof of the proposition with

$$\delta_n = 2 \sum_{k=1}^{k_{max}} 2^k \exp(-2^{k(\beta-2)-1}) \text{ and } k_{max} = \left\lceil \frac{\log_2(n(2^\beta - 1) + 1)}{\beta} \right\rceil.$$



Consent Form

Purpose of the study: This research is being conducted by Eleni Straitouri, Ander Artola Velasco, Stratis Tsirtsis and Manuel Gomez-Rodriguez at the Max Planck Institute for Software Systems. We are inviting you to participate in this research project because you are a Prolific worker. The purpose of this research project is to observe how different automated decision-support systems affect human decisions in sequential tasks.

Procedures: The study involves playing a series of interactive games. The estimated time to complete this study is at most 20 minutes. In each game, you will be given a grid representing a fictional forest map with an ongoing wildfire. Your objective will be to prevent as many cells in the grid as possible from catching fire. In what follows, you will go through a short tutorial explaining the rules of the game and have to pass a short comprehension check before the main experiment.

Potential Risks and Discomforts: There are no risks above those normally encountered on the internet anticipated from participating in this study.

Potential Benefits: There are no direct benefits from participating in this research. We hope that, in the future, other people might benefit from this study through improved understanding of machine-assisted sequential decision making.

Confidentiality: There is no risk of loss of confidentiality, as all study data will be collected and stored anonymously. All study data will not include any personal data, and will be made publicly available to encourage future research on machine-assisted sequential decision making.

Compensation: You will be compensated pursuant with your agreement with Prolific for your participation. You will be responsible for any taxes assessed on the compensation.

Right to Withdraw and Questions: Your participation in this research is completely voluntary. You may choose not to take part at all. If you decide to participate in this research, you may stop participating at any time. If you decide not to participate in this study or if you stop participating at any time, you will not be penalized or lose any benefits to which you otherwise qualify.

If you decide to stop taking part in the study, if you have questions, concerns, or complaints, please contact the investigator:

Eleni Straitouri
Max Planck Institute for Software Systems Paul-Ehrlich Strasse G 26
67663 Kaiserslautern, Germany
+49 631 9303 8317
estraitouri@mpi-sws.org

I confirm that I am 18 years of age or older, I am **NOT** afflicted with color-blindness, I have read this consent form in its entirety or had it read to me, and I voluntarily consent to participate in this research.

Do you agree with the terms of the experiment as explained above? If not please return your submission.

I agree

Figure 7: The consent form that Prolific workers had to read and agree to in order to participate in our study.

B ADDITIONAL EXPERIMENTAL DETAILS

Further details on human subject study. Participation in our study was open to all international Prolific workers independently of their country of residence or their socioeconomic background. Since different countries have different levels of minimum wage, we chose to follow Prolific's payment principles on ethical rewards, which recommend a minimum of 9 GBP per hour for the payment of participants. In addition to that, we decided to provide a bonus payment to the top performing participants to further incentivize participants to pay attention to the game and perform their best. Each participant in our human subject study provided their consent by filling the form shown in Figure 7. After filling this form, each participant went through the tutorial shown in Figure 8, followed by a comprehension check shown in Figure 9. The pool of participants we recruited is characterized by the following demographics: 55% female and 45% male individuals; 5% Asian, 52% African American, 6% Multiracial, 36% White, and 1% with race undisclosed or other; 76% native English speakers; individuals with average age 36.72 with standard deviation 12.39, and median age 32.

Fire spread dynamics. To allow for stochastic fire spread dynamics, we assign a density parameter $p_{i,j}$ to each grid tile with coordinates (i, j), which controls how easy it is for the tile to catch fire and remains constant

Figure 8: Snapshots of the tutorial that the human subject study participants had to go through to understand and familiarize with the game procedures. The snapshots are presented in their order of appearance in the tutorial from left to right and top to bottom.

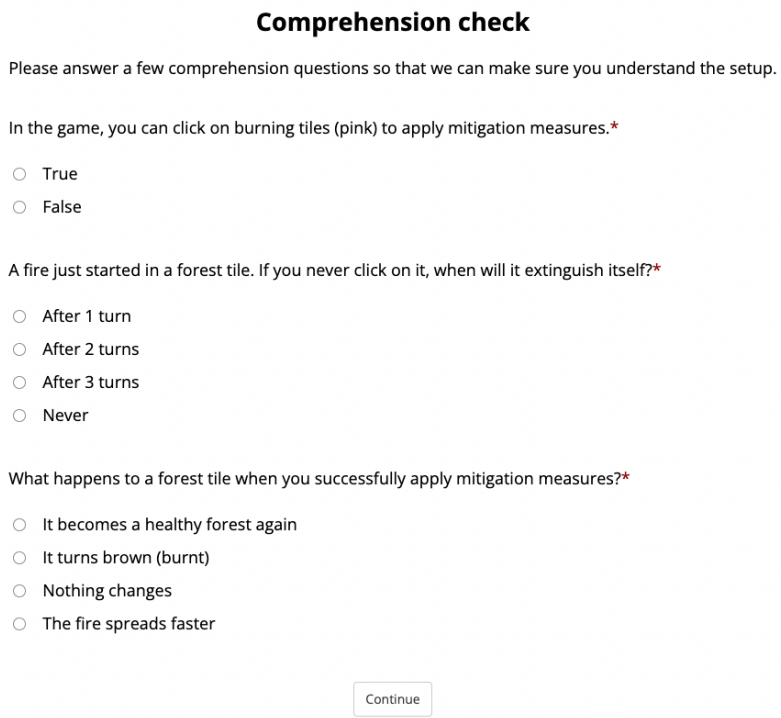


Figure 9: The comprehension test that participants had to pass after the tutorial to proceed with our study.

throughout the game. Each of these parameters takes a value $p_{i,j} \in \{0.1, 0.2, \ldots, 0.9\}$, which we generate using a Gaussian random field, and we visualize using $1, 2, \ldots, 9$ trees, respectively, in the corresponding tile (see Fig. 2).

At each time step t, we decide whether to set the status of each `healthy` tile (i, j) to `burning` proportionally to its density $p_{i,j}$ and the number $|\mathcal{N}_{i,j,t}|$, where $\mathcal{N}_{i,j,t}$ is the set of all `burning` tiles adjacent to (i, j) at time t. Specifically, we consider that each tile $(i', j') \in \mathcal{N}_{i,j,t}$ makes an attempt to spread the fire to the tile (i, j) with probability $p_{i,j}$, and the tile (i, j) catches fire if at least one of these attempts is successful. Formally, we sample a random variable $X_{i,j,t} \sim \text{Binomial}(|\mathcal{N}_{i,j,t}|, p_{i,j})$ and set the status of tile (i, j) to `burning` if $X_{i,j,t} \geq 1$.

Deep Q-Network. We build our AI Agent using reinforcement learning and, during training, we optimize it to operate autonomously in our wildfire mitigation environment without any human in the loop. Specifically, we train a Deep Q-Network (DQN) (Mnih et al., 2015) with a double Q-learning approach (Van Hasselt et al., 2016) to provide valuations $q(\mathbf{s}, a)$ such that they approximate the expected (discounted) cumulative reward the AI agent can obtain by taking action a while seeing a representation $\mathbf{s} = f_S(\mathbf{z})$ and following the optimal policy π^*_M thereafter. To align the AI agent's objective with that of the participants of the human subject study (*i.e.*, saving as many forest tiles as possible), we set the AI agent's (training) reward at each time step t and discount factor γ in a similar manner as in human subject study—we set the reward equal to the negative number of tiles that turned from `healthy` to `burning` in each time step and $\gamma = 0.99$. In addition, we provide a large negative reward equal to the negative total number of `burnt` tiles at the end of each game to speed up training.

To better enable the AI agent to leverage the spatial structure of the environment, we use a network architecture based solely on convolutional layers—four convolutional layers with kernel sizes of 3×3, 3×3, 5×5, and 7×7, respectively, and 32, 32, 64, and 64 filters, respectively, followed by ReLU activations. In addition, to speed up training, we use action masking to ensure that the AI agent only applies water mitigation measures to tiles on the firefront, that is, tiles that are `burning` and have at least one `healthy` neighboring tile.

For training the DQN, we generate 2,000 random game instances and perform 75 simulations for each instance, resulting in a total of 150,000 training games. To ensure that the game instances used for training are sufficiently diverse in terms of their difficulty, we first evaluate the performance achieved by AI agents implementing several heuristic policies (see below) on a large set of randomly generated game instances by measuring the percentage of forest tiles that they manage to maintain `healthy` at the end of the game. Then, we categorize the game

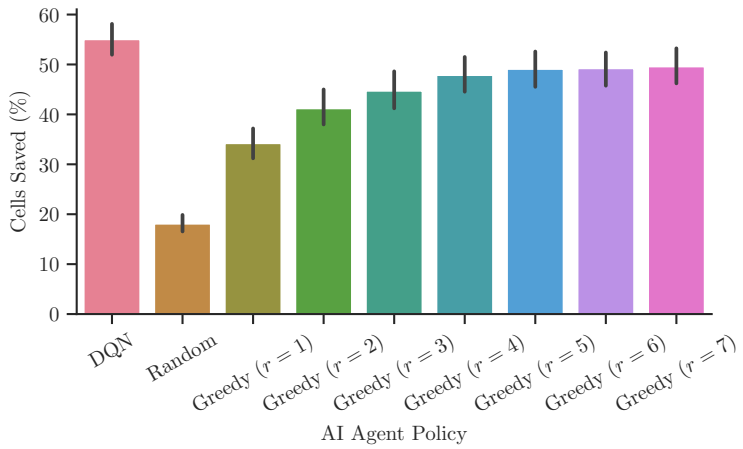


Figure 10: Comparative performance of the trained DQN agent against AI agents implementing heuristic policies.

instances into five difficulty levels based on the performance range (*e.g.*, $20\% - 40\%$) of the best heuristic policy (Greedy with radius $r = 7$, see below), and we create a balanced training set by keeping 400 game instances from each difficulty level. We repeat the same procedure to create a balanced test set of 200 game instances, which we use to evaluate the performance of our AI agent using the trained DQN against AI agents using heuristic policies. Figure 10 shows their comparative performance on the test set, and a detailed description of the heuristic policies can be found below:

- **Random policy**: The agent randomly selects a tile that is on the firefront and applies a water mitigation measure to it.
- **Greedy policy with radius** $r \in \{1, 2, \ldots, 7\}$: For each tile (i, j) on the firefront, the policy computes the product of densities on each path of length r starting from (i, j) and ending in an `healthy` tile (i', j') and adds them up. The agent then applies water mitigation measures to the tile (i, j) that maximizes this sum. In the special case of $r = 1$, the agent simply selects the tile on the firefront whose `healthy` neighbors have the highest total density.

Decision support policies. We select the σ and ε values we use in our human subject study based on simulations we conducted prior to the human subject study. Specifically, we evaluated our decision support policies for ε values from 0.0 to 1.0 with step 0.01 and $\sigma \in \{0.001, 0.01, 0.05, 0.1\}$ using simulated humans, which we elaborate on in further detail in the next paragraph. Based on the simulation results, we identified that our simulated humans under the assistance of our decision support policies achieved the highest average cumulative reward for $\varepsilon \in [0, 0.3]$ and $\sigma = 0.01$. In our human subject study presented in Section 4, we deploy our decision support policies for each ε value in the aforementioned range with step 0.01 and for 10 more values evenly spread outside this range (*i.e.*, in $[0.3, 1.0]$) for completeness.

To build our simulated humans, we conducted a small-scale pilot study prior to our large-scale human subject study. Therein, we asked 30 volunteers to play 200 instances of our wildfire mitigation game on their own (*i.e.*, without any AI assistance). For each set of instances starting from the same state, we measured the average (discounted) cumulative reward achieved by the volunteers and that achieved by AI agents implementing a set of heuristic policies including the greedy policies as described above, as well as softmax versions of the respective greedy policy.[11] Then, we build our simulated humans such that, for each set of instances starting from the same state, they play the game on their own by implementing the heuristic policy that most closely matched the respective average (discounted) cumulative reward achieved by the volunteers in that set of instances. To set the

[11] A softmax heuristic policy with radius r computes the same sum as the respective greedy policy for each tile (i, j) on the firefront and applies a softmax function over those sums. The agent then selects which tile they will apply water mitigation measures on by drawing a random sample from the categorical distribution induced by the values of this softmax function.

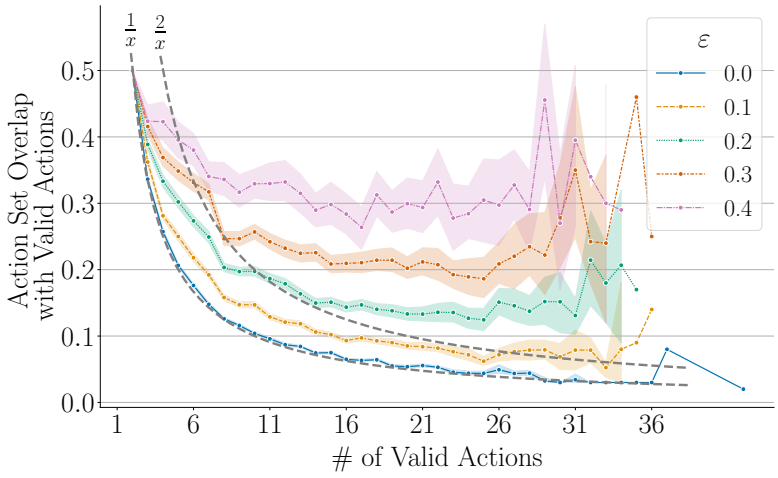


Figure 11: Average action set overlap with valid actions against the number of valid actions. Dashed lines correspond to the action set overlap for action sets of fixed size 1 and 2 respectively. The action set overlap for 1 valid action is always 1.0 and thus omitted to improve visibility. Shaded areas represent 95% confidence intervals.

human policy π_H that our simulated humans follow in the case where they play games under the assistance of our decision support policies, we consider that they implement the same heuristic policy in the respective set of instances, with the only difference that the respective random sampling or argmax operation that leads to the action selection is taken over the (restricted) set of tiles in the action set given by the decision support policy at each time step, rather than over the entire firefront.

Licenses and implementation details. We will publicly release our code and human subject study data under the Creative Commons Attribution License 4.0 (CC BY 4.0). To implement our algorithm and execute our experiments, we use `Python` 3.11.2 along with the open-source libraries `PyTorch` 2.8.0 (Modified BSD License), `NumPy` 2.3.2 (BSD License), `Pandas` 2.3.2 (BSD License). For our human subject study we develop a web application, where we implement the fronend with the `jsPsych` framework (MIT License), and the backend with the open-source libraries `FastAPI` 0.115.12 (MIT License) and `Uvicorn` 0.34.3 (BSD License).

C ADDITIONAL EXPERIMENTAL RESULTS

In this section, we provide additional results on the relative size of action sets returned by our decision support system with respect to the total number of actions one can take at a certain time step (valid actions), *i.e.*, the total number of `burning` tiles at each time step. We compute the relative size at a given time step as the ratio of the number of actions in the action set over the total number of valid actions, to which we refer as the action set overlap with the valid actions. Figure 11 shows that under the optimal ε value, the action set overlap is clearly above the overlap of action sets of fixed size 1—that is the action set overlap achieved by the AI agent alone—thus, further supporting that our decision support system achieves human-AI complementarity. In addition, Figure 11 shows that under the optimal ε value the action set overlap is typically below the overlap of action sets with fixed size 2 and near 0.1 whenever the number of valid actions exceeds 10, *i.e.*, action sets include 10% of the total valid actions whenever the set of valid actions becomes relatively large; Figure 12 suggests that the set of valid actions includes typically up to 20 valid actions.

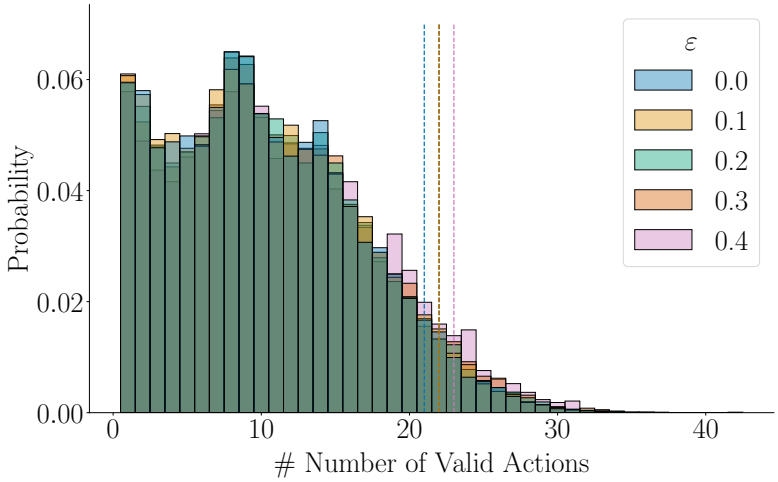


Figure 12: Distributions of valid actions for several bins of ε as in Figure 3. Dashed lines correspond to the 95% empirical quantiles of valid actions. For all shown ε bins the 95% quantile values are in $\{21, 22, 23\}$. To improve visibility we compute the distribution of valid actions after the first time step of each game, as all games begin with 4 valid actions by construction.