Fatigue and Task Load Dependent Decision Referrals for Joint Binary Classification in Human-Automation Teams

Raihan Seraj, *Graduate Student Member, IEEE*, Jerome Le Ny, *Senior Member, IEEE*, and Aditya Mahajan, *Senior Member, IEEE*

***Abstract*—We consider a human-automation team jointly solving binary classification tasks over multiple time stages. At each stage, the automation observes the data for a batch of classification tasks, classifies a subset of them and refers the others to the human. The human's performance depends on task load and fatigue, where fatigue is modeled as a controlled Markov process dependent on the past task loads. We formulate the automation's problem of deciding which tasks to refer as a Markov decision process and present a sampling-based approximate dynamic program that leverages task independence across time and the structure of the recently obtained single-stage optimal allocation policy. We then present a numerical study comparing our solution against a baseline policy that does not explicitly account for fatigue dynamics.**

***Index Terms*—Human-in-the-loop control, Markov processes, stochastic optimal control.**

I. Introduction

HUMAN-AUTOMATION teaming is prevalent in aviation, driving assistance, healthcare, etc., where automation can help human operators sustain task performance over long periods. In such settings, the accumulation of mental fatigue reduces the operator's cognitive flexibility and situational awareness, potentially compromising system performance and safety [1]. Unlike physical fatigue, mental fatigue manifests itself subtly, increasing the risk of errors or lapses in judgment [2].

We consider the design of a decision referral system [3] for human-automation teams performing binary classification tasks, where the automation can refer some of the tasks to the human operator, and classifies the rest. For example, a radiologist and an automation may collaborate to classify medical images (e.g., MRI scans) as "normal" or "abnormal" [4]. The automation can process images quickly and flag ambiguous cases for human evaluation. However, operator fatigue accumulates over time [5], eventually resulting in a higher risk of errors, and hence it is crucial to consider the impact of fatigue when determining which tasks, and how many, to refer to the human. Moreover, instead of relying solely on the immediate fatigue level, an adaptive system that considers the dynamic nature of fatigue over time can improve collaboration and decision-making efficiency.

Task allocation in human-automation teams, shaped by workload, cognitive state, and system performance, is studied for instance in [6], [7], [8]. The adaptive strategies in [8] consider the impact of workload and trust dynamics on the human's decision performance and willingness to follow the automation's recommendations. In [7], the automation takes the cognitive load of the human operator into account to optimally allocate the tasks of an autonomous robot. An adaptive system switching tasks between automated and manual modes based on the operator's electroencephalographic signals is studied in [6]. Cognitive models, such as ACT-R, are used in [9], [10] for dynamic task reallocation by monitoring workload and performance, helping predict operator behavior and evaluate task allocation alternatives.

Some studies consider tasks buffered in a queue and develop policies to dispatch these tasks sequentially to humans with a utilization-dependent service rate [11], to stabilize the queues [12] and optimize attention allocation [13]. However, they abstract the nature of the tasks and do not consider joint decision-making aspects. The papers [3], [14] propose decision referral strategies for binary classification tasks, when increasing task load degrades operator performance. However, they do not consider the dynamics of cognitive states like fatigue. Here we present a decision referral system assuming a given dynamic model of fatigue. Such models can be identified through studies that measure fatigue evolution with cognitive performance [15], through physiological measurements (e.g., heart rate, brain activity, eye movement) [16], [17] or through subjective questionnaires [18].

Our first contribution is to formulate the fatigue-aware decision referral problem as a Markov decision process (MDP). Second, we propose an Approximate Dynamic Program (ADP) to solve the MDP, exploiting the structure of the optimization problem to simplify the computations. Finally, we evaluate

Received 17 March 2025; revised 19 May 2025; accepted 2 June 2025. Date of publication 20 June 2025; date of current version 11 July 2025. This work was supported in part by the Canadian Department of National Defence through the Innovation for Defence Excellence and Security (IDEaS) Micro-Net Program under Grant CFPMN2037, and in part by Fonds de recherche du Québec Ű Nature et technologies (FRQ-NT) under Grant 343486. Recommended by Senior Editor K. Savla. *(Corresponding author: Raihan Seraj.)*

Raihan Seraj and Aditya Mahajan are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC H3A 0E9, Canada (e-mail: raihan.seraj@mail.mcgill.ca; aditya.mahajan@mcgill.ca).

Jerome Le Ny is with the Department of Electrical Enginieering, Polytechnique Montreal, Montreal, QC H3C 3A7, Canada, and also with GERAD, Montreal, QC H3T 1J4, Canada (e-mail: jerome.le-ny@polymtl.ca).

Digital Object Identifier 10.1109/LCSYS.2025.3581647

and compare the performance of our policy against a policy that bases its decisions solely on instantaneous fatigue levels, demonstrating the importance of taking fatigue dynamics into account. In the next section we present the system model. In Section III, we provide details on the dynamic programming solution and our proposed ADP approach. We discuss numerical simulation results in Section IV and conclude in Section V.

Notation: We use $\mathbb{Z}$ and $\mathbb{R}$ to denote the sets of integers and real numbers. For $M \in \mathbb{Z}$ with $M \geq 1$, $[M]$ denotes the set $\{1, \ldots, M\}$. The cardinality of a set $\mathcal{S}$ is denoted $|\mathcal{S}|$.

II. System Model and Problem Formulation

In this section we present a model for a human-automation team jointly solving binary classification tasks over multiple stages. We build on the model presented in [3] for a human-automation team solving a single batch of binary classification tasks. We extend that model to work over multiple stages and include the human's fatigue state into the model.

Consider a human-automation team jointly performing successive batches of binary classification tasks over a finite horizon T. At each time $t \in [T]$, a batch consists of K independent and identically distributed (i.i.d.) tasks, indexed by $k \in [K]$. At time t, each task $k \in [K]$ has an unknown binary state $H_{t,k} \in \{\mathcal{H}_0, \mathcal{H}_1\}$, where $\{H_{t,k}\}_{t\in[T],k\in[K]}$ are independent across tasks and time. The prior probability on the state of each task $k \in [K]$ at time $t \in [T]$ is known, given by $\pi_i = P(H_{t,k} = \mathcal{H}_i), i \in \{0, 1\}$, and is the same across all tasks and time. For each task $k \in [K]$ at time t, the automation and the human receive random observations $Y^a_{t,k} \in \mathcal{Y}^a$ and $Y^h_{t,k} \in \mathcal{Y}^h$ respectively, which depend on the true state $H_{t,k}$ and are assumed conditionally independent given $H_{t,k}$. Moreover, for $m \in \{a, h\}$, the conditional distribution $Y^m_{t,k}$ given $H_{t,k}$ is the same for each $k \in [K]$ and $t \in [T]$. For ease of notation, we use Y^a_t and Y^h_t to denote the random vectors $Y^a_{t,1:K}$ and $Y^h_{t,1:K}$ associated with a batch of tasks.

In addition to the observations for each task, the automation also observes the mental fatigue state[1] of the human operator at each time t, denoted by $F_t \in \mathcal{F}$, where $\mathcal{F}$ is the set of possible fatigue states. This observation could be provided explicitly by the human for instance using the instantaneous self assessment (ISA) [19] via the interface, or inferred from physiological measurements. We assume that the human fatigue state evolves in a controlled Markovian manner with a general functional representation given by

$$F_{t+1} = \phi(F_t, w_t, \eta_t), \tag{1}$$

where w_t is the task load (i.e., the number of tasks referred to the human) at time t and $\{\eta_t\}_{t\geq 1}$ is a noise process, which is i.i.d. across time, and independent of the processes $\{Y^a_t\}_{t\geq 1}$, $\{Y^h_t\}_{t\geq 1}$, and $\{H_{t,1:K}\}_{t\geq 1}$.

The system operates as follows. At time t, the automation observes the data Y^a_t for the entire batch of tasks, as well as the human's state F_t. It then refers a subset of the tasks to the human and classifies the rest as $\mathcal{H}_0$ or $\mathcal{H}_1$. Let $\mathcal{N}_t \subseteq [K]$ denote the indices of the tasks referred to the human (note that $w_t = |\mathcal{N}_t|$). For each task $n \in \mathcal{N}_t$ the human makes an observation $Y^h_{t,n}$ and classifies it as $\mathcal{H}_0$ or $\mathcal{H}_1$.

[1] The methodology generalizes to cognitive states other than fatigue, as long as we can identify the required dynamic and performance models.

A. Example of Fatigue Model

Research shows that sustained cognitive engagement under high task load increases mental fatigue [1]. The following example of dynamic fatigue model is a special case of the general functional representation in (1), where fatigue grows with task load. Suppose $\mathcal{F} = \{0, 1, 2, 3\}$ and

$$\mathbb{P}(F_{t+1} = j \mid F_t = i, |\mathcal{N}_t| = w_t) = [\Lambda(B(w_t))]_{ij}$$

where $B(w) = \lceil 4w/K \rceil$ and $\Lambda(\cdot)$ are transition matrices

$$\Lambda(1) = \begin{bmatrix} 0.8 & 0.2 & 0 & 0 \\ 0.3 & 0.5 & 0.2 & 0 \\ 0 & 0.3 & 0.5 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \Lambda(2) = \begin{bmatrix} 0.4 & 0.6 & 0 & 0 \\ 0.1 & 0.3 & 0.6 & 0 \\ 0 & 0.1 & 0.3 & 0.6 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\Lambda(3) = \begin{bmatrix} 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0.7 & 0.3 \\ 0 & 0 & 0.8 & 0.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad \Lambda(4) = \begin{bmatrix} 0 & 0.4 & 0.6 & 0 \\ 0 & 0 & 0.4 & 0.6 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Here, B is a weakly increasing function and as $B(w_t)$ increases, the probability of transitioning to higher fatigue states increases. This model includes fatigue recovery, where lower task loads (when $B(w_t) = 1$ or 2) allow for a chance of transitioning to a lower fatigue state. On the other hand, the fatigue state $F_t = 3$ is an absorbing state, from which fatigue recovery does not occur [5].

B. Operator Performance With Task Load and Fatigue

We assume that the human classifies a task $n \in \mathcal{N}_t$ based solely on the observation $Y^h_{t,n}$, without accounting for the potentially informative fact that the automation decided to refer this task. This assumption can be justified in practice by the operator's limited decision time and incomplete knowledge of the automation's design rule. We consider that the performance of the human changes with task load w_t [3] and fatigue state F_t [2]. The classification performance of the human is characterized by their true positive $P^h_{tp}(F_t, w_t)$ and false positive $P^h_{fp}(F_t, w_t)$ probabilities, which represent the likelihood of correct and incorrect positive class predictions. These probabilities can be derived from an observation and decision-making model for the human [3], or more conveniently can be estimated through calibration experiments.

An example of operator performance model is

$$\begin{aligned} P^h_{tp}(f, w) &= \max\big(1 - (\alpha_{tp} f + \beta_{tp} w), 0\big), \\ P^h_{fp}(f, w) &= \min\big(\alpha_{fp} f + \beta_{fp} w, 1\big), \end{aligned} \tag{2}$$

with probabilities piecewise linear in the fatigue level f and task load w. Here $(\alpha_{tp}, \alpha_{fp})$ and (β_{tp}, β_{fp}) are positive constants capturing the influence of fatigue and task load.

C. Observation Model of the Automation

We assume that the automation knows its observation model, i.e., the distribution P^a of its observations given $\mathcal{H}_0$ and $\mathcal{H}_1$. Then the joint probability of its observations is

$$\mathbb{P}(Y^a_{t,1:K}) = \prod_{k\in[K]} \sum_{i\in\{0,1\}} \pi_i P^a(Y^a_{t,k} \mid H_{t,k} = \mathcal{H}_i).$$

As an example, consider the following model where the observations of the automation follow Gaussian distributions

$$\begin{aligned} &\text{under } \mathcal{H}_0, \quad Y^a_{t,k} \sim \mathcal{N}(0, \sigma_a^2), \\ &\text{under } \mathcal{H}_1, \quad Y^a_{t,k} \sim \mathcal{N}(d_0, \sigma_a^2), \end{aligned}$$

where the means $0, d_0 > 0$ and variance σ_a^2 are known.

D. Cost and Performance

Let $D_{t,k} \in \{\mathcal{H}_0, \mathcal{H}_1\}$ be the final classification decision made either by the human or the automation for task k at time t. Let $C(D, H)$ denote the cost associated with the classification decision D for a task in true state H. We define

$$\begin{aligned} c_{tp} &:= C(\mathcal{H}_1, \mathcal{H}_1), \quad c_{fp} := C(\mathcal{H}_1, \mathcal{H}_0), \\ c_{tn} &:= C(\mathcal{H}_0, \mathcal{H}_0), \quad c_{fn} := C(\mathcal{H}_0, \mathcal{H}_1). \end{aligned}$$

In addition, the system may incur a cost c_r for each task referred to the human. After the automation observes the entire batch of tasks Y_t^a at time t, the total expected cost from the point of view of the automation depends on the vector of posterior beliefs $p^a_{t,1:K}$ with components given by

$$p^a_{i,t,k} = \mathbb{P}(H_{t,k} = \mathcal{H}_i \mid Y^a_{t,k}), \quad i \in \{0, 1\},\ k \in [K],\ t \in [T]. \tag{3}$$

Therefore, if the automation decides to refer the set $\mathcal{N}_t \subseteq [K]$ and makes a classification decision $D_{t,k}$ on tasks $k \in [K] \setminus \mathcal{N}_t$ and the human makes a classification decision $D_{t,n}$ on tasks $n \in \mathcal{N}_t$ referred to them, the expected per-step classification cost incurred by the system is given by

$$\begin{aligned} \tilde{c}_t(F_t, Y_t^a, \mathcal{N}_t, D_{t,1:K}) = &\sum_{k \in [K] \setminus \mathcal{N}_t} \sum_{i \in \{0,1\}} p^a_{i,t,k} C(D_{t,k}, \mathcal{H}_i) \\ &+ |\mathcal{N}_t| c_r + \sum_{n \in \mathcal{N}_t} \\ &\sum_{i \in \{0,1\}} p^a_{i,t,n} C(D_{t,n}, \mathcal{H}_i). \end{aligned} \tag{4}$$

For tasks $k \in [K] \setminus \mathcal{N}_t$ that are classified by the automation, it selects decisions $D_{t,k}$ to minimize the expected cost given by the first term of (4), which can be rewritten as

$$C^a(Y^a_{t,k}) := \min\{C_0(Y^a_{t,k}), C_1(Y^a_{t,k})\}$$

where

$$\begin{aligned} C_0(Y^a_{t,k}) &= (1 - p^a_{1,t,k}) c_{tn} + p^a_{1,t,k} c_{fn} \quad (\text{for } D_{t,k} = \mathcal{H}_0), \\ C_1(Y^a_{t,k}) &= (1 - p^a_{1,t,k}) c_{fp} + p^a_{1,t,k} c_{tp} \quad (\text{for } D_{t,k} = \mathcal{H}_1). \end{aligned}$$

For tasks $n \in \mathcal{N}_t$, referred to the human at time t, the automation does not know the final decisions of the human $\{D_{t,n}\}_{n \in \mathcal{N}_t}$, and forms a posterior belief on them. Therefore, the expectation of (4) averaged over the decisions of the human can be written as

$$\begin{aligned} &c_t(F_t, Y_t^a, \mathcal{N}_t) \\ &= \sum_{k \in [K] \setminus \mathcal{N}_t} C^a(Y^a_{t,k}) + |\mathcal{N}_t| c_r + \sum_{n \in \mathcal{N}_t} \Gamma(F_t, Y^a_{t,n}, |\mathcal{N}_t|), \end{aligned} \tag{5}$$

where

$$\begin{aligned} &\Gamma(F_t, Y^a_{t,n}, |\mathcal{N}_t|) \\ &= (1 - p^a_{1,t,n}) \Big[P^h_{fp}(F_t, |\mathcal{N}_t|) c_{fp} + (1 - P^h_{fp}(F_t, |\mathcal{N}_t|)) c_{tn} \Big] \\ &\quad + p^a_{1,t,n} \Big[P^h_{tp}(F_t, |\mathcal{N}_t|) c_{tp} + (1 - P^h_{tp}(F_t, |\mathcal{N}_t|)) c_{fn} \Big]. \end{aligned}$$

E. Optimization Problem

The automation's decision-making problem is formulated as an MDP as follows. The set of states of the MDP is $\mathcal{F} \times \mathcal{Y}_K^a$, where $\mathcal{Y}_K^a := (\mathcal{Y}^a)^K$, and the state at time t is (F_t, Y_t^a). The set of actions is all the subsets of $[K]$ and the action at time t is $\mathcal{N}_t$. At time t, the system incurs a cost $c_t(F_t, Y_t^a, \mathcal{N}_t)$ given by (5). Given a finite horizon T, we aim to find a control policy $g = (g_1, \dots, g_T)$, where g_t is a control law that selects the action $\mathcal{N}_t$ as a function of F_t and Y_t^a, to minimize the expected total cost

$$J(g) = \mathbb{E}\left[\sum_{t=1}^{T} c_t(F_t, Y_t^a, g_t(F_t, Y_t^a)) \right].$$

III. Dynamic Programming Decomposition

The optimization problem above can be solved using dynamic programming over a finite horizon to determine the value functions, $\{V_t\}_{t=1}^{T+1}$, where the value function $V_t \colon (\mathcal{F} \times \mathcal{Y}^a) \to \mathbb{R}$ represents the minimum expected cumulative cost achievable from time t onward by following the optimal policy. The value functions can be obtained recursively, from $T+1$ backwards, as follows: for $t = T+1$ and all $f \in \mathcal{F}$ and $y^a \in \mathcal{Y}_K^a$, initialize $V_{T+1}(f, y^a) = 0$. Then, for $t \in [T]$, $f \in \mathcal{F}$, and $y^a \in \mathcal{Y}_K^a$, recursively define

$$\begin{aligned} Q_t(f, y^a, n) = &c_t(f, y^a, n) \\ &+ \mathbb{E}\Big[V_{t+1}(F_{t+1}, Y^a_{t+1}) \,\Big|\, F_t = f, \mathcal{N}_t = n \Big] \end{aligned} \tag{6}$$

$$\text{and} \qquad V_t(f, y^a) = \min_{n \subseteq [K]} Q_t(f, y^a, n). \tag{7}$$

Note that in (6), we do not condition on $Y_t^a = y^a$ because the observations are independent across time.

Then, a policy $g = (g_1, \dots, g_T)$ is optimal if and only if for all $t \in [T]$ it satisfies

$$g_t(f, y^a) \in \arg\min_{n \subseteq [K]} Q_t(f, y^a, n), \quad \forall f \in \mathcal{F}, y^a \in \mathcal{Y}_K^a. \tag{8}$$

There are three computational challenges associated with solving the above dynamic program. At each stage: (i) we need to compute the function Q_t for all values of y^a in $\mathcal{Y}_K^a$; (ii) we need to take expectations over Y^a_{t+1} which lies in $\mathcal{Y}_K^a$; and we need to minimize Q_t over all possible subsets of $[K]$. We now present three simplifications to circumvent these computational challenges. First, we use the ideas from [3] and consider the subproblem of optimizing over all possible task loads at each time. Second, we simplify the dynamic program by treating the fatigue state F_t as a post-decision state. And, finally, we present an ADP where the expectation over Y^a_{t+1} is computed via Monte Carlo approximation.

A. Reduction to a Task Load Optimization Problem

The cost function given by (5) can be rewritten as

$$c_t(F_t, Y_t^a, \mathcal{N}_t) = \sum_{k \in [K]} C^a(Y^a_{t,k}) - \sum_{n \in \mathcal{N}_t} R(Y^a_{t,n}, F_t, |\mathcal{N}_t|), \tag{9}$$

where $R(Y^a_{t,n}, F_t, |\mathcal{N}_t|)$ denote the *referral indices*, given by

$$R(Y^a_{t,n}, F_t, |\mathcal{N}_t|) = C^a(Y^a_{t,n}) - \Gamma(F_t, Y^a_{t,n}, |\mathcal{N}_t|) - c_r. \tag{10}$$

Only the second term in (9) depends on $\mathcal{N}_t$. Therefore, for a fixed value of $|\mathcal{N}_t|$, $c_t(F_t, Y_t^a, \mathcal{N}_t)$ is minimized when $\mathcal{N}_t$ is chosen to maximize $\sum_{n \in \mathcal{N}_t} R(Y^a_{t,n}, F_t, |\mathcal{N}_t|)$, i.e., the cost reduction when referring tasks $\mathcal{N}_t$ to the human. Therefore, similarly to [3, Lemma 1], we get the following result.

Proposition 1: For a given task load $|\mathcal{N}_t| = w$ at time t, the instantaneous cost (9) is minimized by referring the w tasks with the highest referral indices.

For a fixed value of $|\mathcal{N}_t|$, let $c_t^*(F_t, Y_t^a, |\mathcal{N}_t|)$ denote the minimum cost,

$$c_t^*(F_t, Y_t^a, |\mathcal{N}_t|) = \min_{\mathcal{N}_t\,:\,|\mathcal{N}_t|=w} \sum_{k\in[K]} C^a(Y_{t,k}^a) - \sum_{n\in\mathcal{N}_t} R(Y_{t,n}^a, F_t, |\mathcal{N}_t|). \quad (11)$$

Note that this cost can be computed efficiently by following Proposition 1. Since the fatigue dynamics (1) depends only on $|\mathcal{N}_t|$, we can rewrite the recursive step (6) and (7) of the dynamic program as follows: for $t \in [T]$, $f \in \mathcal{F}$, and $y^a \in \mathcal{Y}_K^a$,

$$V_t(f, y^a) = \min_{0\le w\le K}\Big\{c_t^*(f, y^a, w) + \mathbb{E}[V_{t+1}(F_{t+1}, Y_{t+1}^a) \mid F_t = f, |\mathcal{N}_t| = w]\Big\}. \quad (12)$$

B. Simplified Value Function in Terms of the Fatigue State

For further computational efficiency, we exploit the fact that the observations Y_t^a are independent across time. For this, define the *simplified value function* $\{\bar{V}_t\}_{t=1}^{T+1}$, $\bar{V}_t : \mathcal{F} \to \mathbb{R}$, in terms of the fatigue state as $\bar{V}_t(f) = \mathbb{E}[V_t(f, Y_t^a)]$. It satisfies $\bar{V}_{T+1}(f) = 0$ for all $f \in \mathcal{F}$, and for $t \in [T]$ and $f \in \mathcal{F}$ and $t \in [T]$

$$\bar{V}_t(f) = \mathbb{E}\Big[\min_{0\le w\le K}\Big\{c^*(f, Y_t^a, w) + \mathbb{E}\big[\bar{V}_{t+1}(F_{t+1}) \mid F_t = f, |\mathcal{N}_t| = w\big]\Big\}\Big]. \quad (13)$$

Computing $\{V_t\}_{t=1}^{T+1}$ using the dynamic program (7) suffers from the curse of dimensionality because $|\mathcal{Y}_K^a| = |\mathcal{Y}^a|^K$. However, with the simplified value function $\{\bar{V}_t\}_{t=1}^{T+1}$, storing the values for all $f \in \mathcal{F}$ suffices.

C. Monte Carlo Approximation of the Expectation Over Observations

To compute the simplified value function using (13) in practice, we approximate the expectation with respect to Y_t^a using Monte Carlo samples. For M a large integer defining the number of Monte Carlo samples, let $\{\widehat{\bar{V}}_t^{(M)}\}_{t=1}^{T+1}$ be an approximation of $\{\bar{V}_t\}_{t=1}^{T+1}$, defined recursively as follows. For all $f \in \mathcal{F}$, $\widehat{\bar{V}}_{T+1}^{(M)}(f) = 0$. Then, for all time $t \in [T]$,

$$\widehat{\bar{V}}_t^{(M)}(f) = \frac{1}{M}\sum_{m=1}^{M}\Big[\min_{0\le w\le K}\Big[c_t^*(f, y^{a,(m)}, w) + \mathbb{E}\big[\widehat{\bar{V}}_{t+1}^{(M)}(F_{t+1}) \mid F_t = f, w\big]\Big]\Big], \quad (14)$$

where $\{y^{a,(m)}\}_{m=1}^{M}$ are i.i.d. samples of Y_t^a. We next show that this randomized approximation converges to the desired value function $\bar{V}_t$, at least under some technical assumption on the state-space $\mathcal{F}$.

Theorem 1: Assume $\mathcal{F}$ is countable. For all time $t \in [T+1]$, almost surely the sequence $\widehat{\bar{V}}_t^{(M)}$ converges to $\bar{V}_t$ pointwise, i.e., $\mathbb{P}(\forall f \in \mathcal{F}, \lim_{M\to\infty} \widehat{\bar{V}}_t^{(M)}(f) = \bar{V}_t(f)) = 1$.

Proof: We proceed by backward induction. The result is trivially true at time $T+1$. Suppose now that it is true at some time index $t+1$, we want to show that it is also true at time index t.

Let Ω be the underlying abstract sample space, and $\Omega_1 \subset \Omega$ be a set of probability 1 over which $\widehat{\bar{V}}_{t+1}^{(M)}$ converges pointwise to $\bar{V}_{t+1}$ as $M \to \infty$. Since the (random) functions $\widehat{\bar{V}}_{t+1}^{(M)}$ are uniformly bounded (e.g., by $T \times \max(c_{fp}, c_{fn}, c_{tp}, c_{tn})$), by the dominated convergence theorem we have, for any $f \in \mathcal{F}$, $w \in \{0, \dots, K\}$, and for any $\omega \in \Omega_1$,

$$\mathbb{E}_\eta\Big[\widehat{\bar{V}}_{t+1}^{(M)}(\phi_t(f, w, \eta))\Big] \xrightarrow[M\to\infty]{} \mathbb{E}_\eta\big[\bar{V}_{t+1}(\phi_t(f, w, \eta))\big]. \quad (15)$$

Define now, for all $f \in \mathcal{F}$, $y \in \mathcal{Y}_K^a$, $w \in \{0, \dots, K\}$,

$$\bar{Q}_t^{(M)}(f, y, w) = c_t^*(f, y, w) + \mathbb{E}_\eta\Big[\widehat{\bar{V}}_{t+1}^{(M)}(\phi_t(f, w, \eta))\Big],$$
$$\bar{Q}_t(f, y, w) = c_t^*(f, y, w) + \mathbb{E}_\eta\big[\bar{V}_{t+1}(\phi_t(f, w, \eta))\big],$$

and note that from (12)

$$\min_{0\le w\le K} \bar{Q}_t(f, y, w) = V_t(f, y),$$
$$\text{so } \mathbb{E}\Big[\min_{0\le w\le K} \bar{Q}_t(f, Y_t^a, w)\Big] = \bar{V}_t(f).$$

Let $f \in \mathcal{F}$. Let $\epsilon > 0$. By (15), there exists some integer M_0 such that for all $M \ge M_0$, we can show that for all $y \in \mathcal{Y}^a$, for all $\omega \in \Omega_1$,

$$\Big|\min_{0\le w\le K} \bar{Q}_t^{(M)}(f, y, w) - \min_{0\le w\le K} \bar{Q}_t(f, y, w)\Big| \le \epsilon.$$

Hence, for all $\omega \in \Omega_1$, for all $M \ge M_0$,

$$\Big|\mathbb{E}\Big[\min_{0\le w\le K} \bar{Q}_t^{(M)}(f, Y_t^a, w)\Big] - \bar{V}_t(f)\Big| \le \epsilon. \quad (16)$$

Now, note from (14) that

$$\widehat{\bar{V}}_t^{(M)}(f) = \frac{1}{M}\sum_{m=1}^{M}\min_{0\le w\le K} \bar{Q}_t^{(M)}(f, Y_t^a, w).$$

By the strong law of large numbers, for all $\omega \in \Omega_1 \setminus N_f$, where N_f has probability 0 (and can depend on f), there exists $M_1 \ge M_0$, such that for all $M \ge M_1$

$$\Big|\widehat{\bar{V}}_t^{(M)}(f) - \mathbb{E}\Big[\min_{0\le w\le K} \bar{Q}_t^{(M)}(f, Y_t^a, w)\Big]\Big| \le \epsilon. \quad (17)$$

Combining (16) and (17), we get that $\widehat{\bar{V}}_t^{(M)}(f) \to \bar{V}_t(f)$ as $M \to \infty$, for all $\omega \in \Omega_1 \setminus N_f$.

Using this argument for any $f \in \mathcal{F}$, we see that $\widehat{\bar{V}}_t^{(M)}$ converges pointwise to $\bar{V}_t$ for all $\omega \in \Omega_1 \setminus (\cup_{f\in\mathcal{F}} N_f)$. This set is still of probability 1 because $\mathcal{F}$ is assumed countable. This completes the induction step and the proof. ■

Note that the convergence result of Theorem 1 may be true under more general assumptions on the space $\mathcal{F}$, for a more general discussion on convergence, refer to the work on empirical dynamic programming in [20].

As was the case for $\bar{V}_t$, the main advantage of $\widehat{\bar{V}}_t^{(M)}$ is that we do not need to pre-compute the optimal for all $y^a \in \mathcal{Y}_k^a$. For the specific realization of $y^a \in \mathcal{Y}_k^a$ observed at time t, we can compute an approximation $\widehat{V}_t^{(M)}$ of $\widehat{V}_t$ as follows:

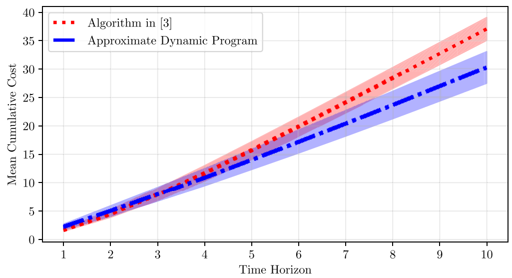

Fig. 1. Comparison of mean cumulative costs for different time horizons, computed from 1000 simulations. The bands indicate one standard deviation on each side.

$$\widehat{V}_t^{(M)}(f, y^a) = \min_{0 \le w \le K} \Big[c_t^*(f, y^a, w) + \mathbb{E}[\widehat{\widetilde{V}}_{t+1}^{(M)}(F_{t+1}) \mid F_t = f, |\mathcal{N}_t| = w] \Big].$$

Consequently we can compute an approximately optimal policy $\hat{g}_t^{(M)}$ as follows

$$\hat{g}_t^{(M)}(f, y^a) \in \arg\min_{0 \le w \le K} \Big[c_t^*(f, y^a, w) + \mathbb{E}[\widehat{\widetilde{V}}_{t+1}^{(M)}(F_{t+1}) \mid F_t = f, |\mathcal{N}_t| = w] \Big].$$

Therefore our approximate dynamic program (ADP) constitutes of recursively computing the approximate value function $\widehat{V}_t^{(M)}$ and obtaining an approximately optimal policy $\hat{g}_t^{(M)}$.

IV. Simulation Results

We now present simulation results for our ADP method, where we optimize the decision referral policy considering fatigue dynamics. For our numerical experiments, we use $T = 10$, $K = 20$, $M = 1000$ samples, $\pi = [0.5, 0.5]$, $c_{tp} = c_{tn} = 0$, $c_{fp} = c_{fn} = 1$, $c_r = 0$. We use the models from Section II-A, Section II-B with parameters $(\alpha_{tp}, \alpha_{fp}) = (0.087, 0.1)$ and $(\beta_{tp}, \beta_{fp}) = (0.043, 0.033)$, and Section II-C with $(d_0, \sigma_a) = (3, 2.3)$. We compare the ADP solution with the algorithm proposed in [3], which follows a myopic policy that optimizes only the immediate cost at each time step, as described in Proposition 1. This approach does not account for the impact of actions on future fatigue levels, as it does not consider any model of fatigue dynamics.

A. Comparison of the Mean Cumulative Cost

We run Monte Carlo simulations over 1000 independent sample paths to evaluate the performance of the policy obtained by ADP algorithm and the algorithm in [3] as a function of the horizon, starting with initial state $F_1 = 0$.

The mean cumulative performance as a function of horizon is shown in Figure 1. For a short horizon, these mean cumulative costs are roughly the same. However, as the time horizon increases, the ADP algorithm provides significant cost improvements.

To understand the difference in performance of the two algorithms, in Figure 2, we compare the cost incurred by the automation, cost incurred by the human, and the total cost for the horizon T. The plots show that ADP provides significantly lower cost for the human and only marginally higher cost for the automation, helping the fatigue-aware decision referral policy achieve better overall team performance.

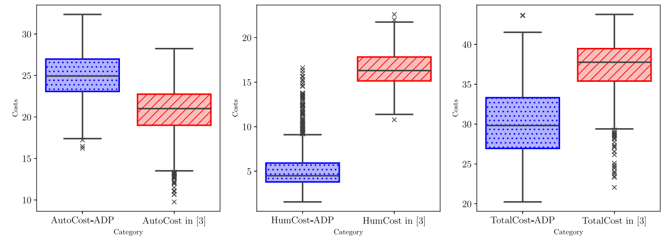

Fig. 2. Comparison of automation, human and total cost. The markers "×" represent outlier values (i.e., outside of $[Q_1 - 1.5(Q_3 - Q_1), Q3 + 1.5(Q_3 - Q1)]$ where Q_1 and Q_3 are the 25th and 75th percentiles).

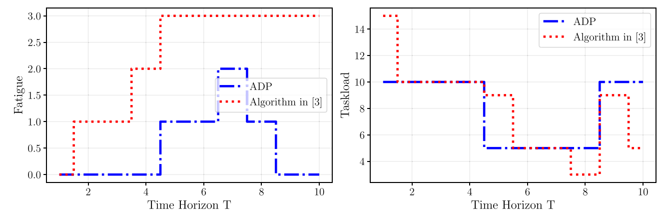

Fig. 3. A sample path of fatigue and taskload evolution with ADP and algorithm in [3] for low initial fatigue state.

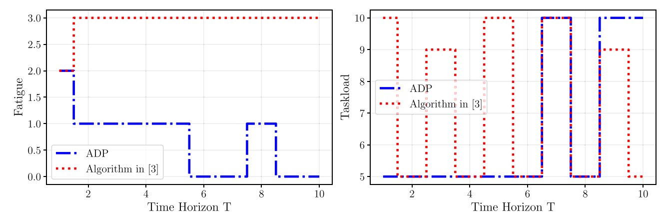

Fig. 4. Fatigue and Taskload evolution with ADP and algorithm in [3] for high initial fatigue state.

B. Comparison of Sample Path Behaviors

To understand the behavioral difference between the two algorithms, in Figures 3 and 4 we compare their sample path behavior for a single realization starting at low $(F_1 = 0)$ and high $(F_1 = 2)$ initial fatigue states, respectively. Figure 3 shows that the ADP solution adjusts task load to maintain fatigue at an appropriate level, preventing it from reaching the absorbing state $(F_t = 3)$ where recovery is not possible. In contrast, the algorithm in [3] assigns higher task load, pushing fatigue into the absorbing state. Figure 4 shows that with high initial fatigue, ADP anticipates the increase in future fatigue and reduces task load to allow for recovery, unlike the algorithm in [3] which assigns task load based on current fatigue and risks reaching the absorbing high fatigue state, failing to achieve recovery. We can see that in this example, under the ADP policy fatigue builds up more slowly and the automation assigns fewer tasks to the human operator.

C. Robustness to Model Uncertainty

In practice, the dynamics of the fatigue model would be learned via calibration experiments. Therefore, it is important to understand the robustness of the performance to uncertainty in model dynamics. For that matter, we compute the ADP policy for the nominal fatigue model of Section II-A and evaluate it on different, randomly sampled models in an ε-ball $(\varepsilon = 0.05)$ around the nominal model. We sample five such systems denoted by (System-1,…,System-5) where for each system we perturb each element $\Lambda_{i,j}$ of the transition matrix as $\widetilde{\Lambda}_{i,j} = \max(\Lambda_{i,j} + \delta, 0) / \sum_j (\max(\Lambda_{i,j} + \delta, 0))$, where $\delta \sim$

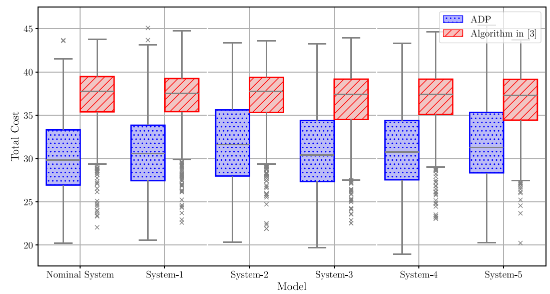

Fig. 5. Comparison of the performance of the ADP and the algorithm in [3] for different perturbed fatigue models. The markers "×" represent outlier values.

$\mathtt{Unif}(-\varepsilon, \varepsilon)$. We run Monte Carlo simulation over 1000 sample paths to evaluate the performance of the nominal policy on the five perturbed models and compare them with performance of algorithm in [3], starting with initial state $F_1 = 0$. The results, shown in Figure 5, show that for all the five perturbed systems, the ADP policy for the nominal model consistently achieves lower total cost compared to the algorithm in [3]. This suggests that the ADP policy shows some level of robustness to errors in the identification of the dynamic fatigue model.

V. Conclusion

The evolution of cognitive fatigue and other mental states impacts the performance of human-automation teams. In this letter, we develop a strategy for decision referrals by an automated system to a human operator, and show that by taking fatigue dynamics into account, one can significantly improve overall decision accuracy. Optimizing the decision referral policy can be posed as a Markov decision process, albeit with a very large state space accounting for all the possible observations that the automation can make. Hence, we introduce a significantly more efficient approximate dynamic programming (ADP) methodology, able to compute the decision referral strategy for the specific observations made at each period, rather than requiring to compute and store the value function for all possible observations.

In this letter, we assume that the automation is provided with a model of fatigue dynamics. In practice, such a model can be estimated using a combination of subjective, physiological and task performance measurements, such as task completion time, accuracy, etc. In future work, we aim to validate our approach through experiments with human participants and estimate such dynamic models from real-world data. Furthermore, we plan to explore reinforcement learning approaches where the automation learns a decision referral strategy by observing fatigue state samples, and extend our methodology to multi-class classification problems.

References

[1] M. Hinss, A. Brock, and R. Roy, "Cognitive effects of prolonged continuous human-machine interaction: The case for mental state-based adaptive interfaces," *Front. Neuroergonomics*, vol. 3, Aug. 2022, Art. no. 935092. [Online]. Available: https://www.frontiersin.org/journals/neuroergonomics/articles/10.3389/fnrgo.2022.935092

[2] C. Brady, S. Sawant, K. C. Madathil, and N. McNeese, "A systematic review on the effect of cognitive fatigue in teams," in *Proc. Human Factors Ergon. Soc. Annu. Meet.*, vol. 68, 2024, pp. 1287–1291.

[3] K. Kaza, J. Le Ny, and A. Mahajan, "Decision referrals in human-automation teams," in *Proc. Conf. Decision Control*, 2021, pp. 2842–2847.

[4] R. Yu et al., "PI-RADSAI: Introducing a new human-in-the-loop AI model for prostate cancer diagnosis based on MRI," *Brit. J. Cancer*, vol. 128, no. 6, pp. 1019–1029, 2023.

[5] J. Matthews et al., "Computational mechanisms underlying the dynamics of physical and cognitive fatigue," *Cognition*, vol. 240, Nov. 2023, Art. no. 105603.

[6] N. R. Bailey, M. W. Scerbo, F. G. Freeman, P. J. Mikulka, and L. A. Scott, "Comparison of a brain-based adaptive system and a manual adaptable system for invoking automation," *Human Factors*, vol. 48, no. 4, pp. 693–709, 2006.

[7] T. R. Giele, T. Mioch, M. A. Neerincx, and J.-J. C. Meyer, "Dynamic task allocation for human-robot teams," in *Proc. Int. Conf. Agents Artif. Intell.*, 2015, pp. 117–124.

[8] C. Dubois and J. Le Ny, "Adaptive task allocation in human-machine teams with trust and workload cognitive models," in *Proc. Int. Conf. Syst. Man, Cybern.*, 2020, pp. 3241–3246.

[9] C. R. Fisher, M. E. Frame, and C. Stevens, "Using cognitive models to design dynamic task allocation systems," *J. Defense Model. Simul.*, vol. 20, no. 4, pp. 467–480, 2023.

[10] M. E. Frame, A. S. Boydstun, A. M. Maresca, and J. S. Lopez, "Development of an autonomous manager for dyadic human-machine teams in an applied multitasking surveillance environment," in *Proc. Int. Conf. Intell. Human Syst. Integr.*, 2019, pp. 706–711.

[11] V. Jog, R. J. La, M. Lin, and N. C. Martins, "Channels, remote estimation and queueing systems with a utilization-dependent component: A unifying survey of recent results," 2019, *arXiv:1905.04362*.

[12] K. Savla and E. Frazzoli, "Maximally stabilizing task release control policy for a dynamical queue," *IEEE Trans. Autom. Control*, vol. 55, no. 11, pp. 2655–2660, Nov. 2010.

[13] V. Srivastava, A. Surana, and F. Bullo, "Adaptive attention allocation in human-robot systems," in *Proc. Amer. Control Conf.*, 2012, pp. 2767–2774.

[14] B. Hyun, P. Kabamba, and A. Girard, "Optimal classification by mixed-initiative nested thresholding," *IEEE Trans. Cybern.*, vol. 45, no. 1, pp. 29–39, Jan. 2015.

[15] K. Kunasegaran, A. M. H. Ismail, S. Ramasamy, J. V. Gnanou, B. A. Caszo, and P. L. Chen, "Understanding mental fatigue and its detection: A comparative analysis of assessments and tools," *PeerJ*, vol. 11, Aug. 2023, Art. no. e15744. [Online]. Available: https://doi.org/10.7717/peerj.15744

[16] T. Bafna and J. P. Hansen, "Mental fatigue measurement using eye metrics: A systematic literature review," *Psychophysiology*, vol. 58, no. 6, 2021, Art. no. e13828. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/psyp.13828

[17] S.-Y. Cheng and H.-T. Hsu, "Mental fatigue measurement using EEG," in *Risk Management Trends*, G. Nota, Ed., London, U.K.: IntechOpen, 2011, ch. 11. [Online]. Available: https://doi.org/10.5772/16376

[18] J. DeLuca, "The assessment and measurement of fatigue," in *Fatigue as a Window to the Brain*. Cambridge, MA, USA: MIT Press, 2005, pp. 19–36. [Online]. Available: https://books.google.ca/books?id=Y2CxzZo3ZvUC

[19] A. Hamann and N. Carstengerdes, *Fatigue Instantaneous Self-Assessment (F-ISA): Development of a Short Mental Fatigue Rating*, Deutsches Zentrum für Luft-und Raumfahrt (DLR), Cologne, Germany, 2020.

[20] W. B. Haskell, R. Jain, and D. Kalathil, "Empirical dynamic programming," *Math. Oper. Res.*, vol. 41, no. 2, pp. 402–429, 2016.