Learning About the Learner: Reinforcement Learning for Fault Recovery Using Gaussian Processes With Contextual Measurements

Steve McGuire*
University of California Santa Cruz, Santa Cruz, CA 95064

P. Michael Furlong†
University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Christoffer Heckman‡
University of Colorado at Boulder, Boulder, CO 80309

Simon Julier§
University College London, London, UK

Nisar Ahmed¶
University of Colorado at Boulder, Boulder, CO 80303

This work addresses the iterated non-stationary assistant selection problem, in which over the course of repeated interactions on a mission, an autonomous robot experiencing a fault must select a single human from among a group of assistants to restore it to operation. The assistants in our problem have a level of performance that changes as a function of their experience solving the problem. Our approach utilizes reinforcement learning via a multi-arm bandit formulation to learn about the capabilities of each potential human assistant and decide which human to task. Building on our past work, we evaluate the potential for a Gaussian process-based machine learning method to effectively model the complex dynamics associated with human learning and forgetting. Application of our method in simulation shows that our method is capable of tracking performance of human-like dynamics for learning and forgetting. Using a novel selection policy called the proficiency window, we show that our technique can outperform baseline selection strategies while providing guarantees on human utilization. Our work offers an effective potential alternative to dedicated human supervisors, with application to any human-robot system where a set of humans is responsible for overseeing autonomous robot operations.

*Assistant Professor, Electrical and Computer Engineering
†Postdoctoral Research Fellow, Centre for Theoretical Neuroscience
‡Assistant Professor, Computer Science
§Reader, Computer Science
¶Assistant Professor, Ann and H.J. Smead Department of Aerospace Engineering Sciences

I. Introduction

Roboticists have yet to create an autonomous robot that unfailingly performs its duties without human intervention. Until robots reach this ideal goal, they must instead plan for failures to occur and robustly return to autonomous operations. This work addresses an open problem in human-machine teaming regarding *fault recovery*; in particular, we are interested in improving how an autonomous robot selects a human assistant to provide aid and return to nominal operating condition. As a framework, we use the idealized decision making feedback structure in Fig. 1, where an assistant selector chooses a human, observes an outcome, and then applies machine learning to improve the selector's future decisions.

As a motivating example, consider an autonomous robot on a mission of long-term space exploration in concert with humans. In general, robots are used in dull, dirty, and dangerous missions; exploration is no different. Exploration offers several unique challenges to the design of an autonomous system that are simply not present under more controlled scenarios such as manufacturing. Prime among these challenges for systems that need to operate for weeks, months, and even years is that of *uncertainty* on the part of the surrounding environment, the attending humans, as well as the autonomous system.

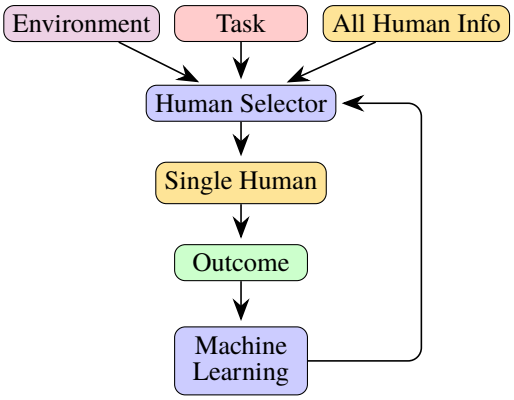


Fig. 1 An idealized selector decision making process incorporates the task, the current environment, information about all potential humans, and a history of past responses.

In a manufacturing environment, operating conditions can be controlled to an exquisite degree, ranging from illumination to humidity. There is very little uncertainty in actions because of the well-known nature of the environment. By the very definition of *exploration*, there are elements of the environment that are not fully known with potentially mission-ending impact. Current space exploration robotics utilize an extremely conservative approach, where operations are choreographed and simulated extensively in order to minimize risk [1]; recent advances in autonomy are slow to be adopted [2].

In the space exploration domain, the humans that are potentially available to assist a robot are highly trained individuals that are already being monitored for health reasons. The current state of the art for a robot that has experienced an autonomy failure involves permanently choosing the same person or operational role to help it every time. This strategy ignores available measurements that can be taken of potential assistants to help the robot make a better choice, as well as the time history of past choices and observed outcomes.

Our approach to address the problems inherent with static selection has consisted of three main themes: in previous work [3, 4], we explored the impact of *indicators of opportunity* (IOOs), *individual modeling*, and *long-term operations*. IOOs are observations of a particular human assistant that potentially inform an assistant's internal hidden cognitive

condition and thus inform potential performance. These IOOs can consist of behavioral observations, such as mouse use habits or other interface interaction, as well as physical observations such as heart rate variability and pupillometry; together, these observations form a *context vector* that describes each human assistant's state. However, the relationships between context variables and expected task performance are complex and intertwined, potentially varying on a per-person basis. Using a reinforcement learning, we can approximate these relationships using online observations. By modeling the assistant selection problem as one of multi-arm bandit reinforcement learning, we have validated in human studies that these observations can be used to predict task performance on an individual basis, recognizing the potential uniqueness of individuals' response to stressful environments. Previously, we had assumed that a human assistant's underlying reward function of their observed IOO was unknown, but fixed. In this work, we present an alternative that removes this constancy assumption and explores our final theme of long-term operations. Specifically, we assess the usefulness of a reward function model that can dynamically adjust to a human's changing response; in short, we have constructed an assistant selection system for fault recovery capable of learning that human assistants are also learning about both the system's failure modes and steps necessary to recover from failure. Although developed in the context of space exploration, our system is widely applicable to many uncertainty-laden problem domains where humans and imperfect autonomous systems work together.

Our work has three major contributions. First, we identify models of human learning from operations literature that are capable of a higher-fidelity modeling of the human task learning process. Second, we present a novel formulation of a multi-arm bandit technique utilizing Gaussian processes (GPs) capable of modeling the type of performance data associated with human learning dynamics while exploiting contextual data. Finally, we assess the usefulness of our GP-based context-aware reinforcement learning technique in a series of truth model simulations to provide intuition to a prospective system designer. As a result of our simulations, we conclude that our GP-based technique is capable of modeling the time and experience-dependent nature of human learning within a reinforcement learning, multi-arm bandit framework and surpassing the performance of competing approaches.

II. Background

Our work is motivated by a conceptual model of an ideal learning robot shown in Fig. 2. This conceptual breakdown is general enough to be applied to many different autonomous robotics scenarios; in particular, we are interested in scenarios where the value of a human's time is so great that assigning a human to supervise the robot as a dedicated duty would be unacceptable. Such scenarios could include future manned space exploration missions [5] or present-day operations such as supervision of autonomous ground passenger or cargo vehicles. Autonomy failures may take a wide variety of forms, from 'hard' failures such as seized motors or unresponsive electronics, to 'soft' failures such as a navigation system that was unable to reach its desired end goal. Our work focuses on soft failures, where a robot's inability to complete an autonomous task in no way has the potential for vehicle loss or end-of-mission.

In our model, an external *planner* decomposes high-level mission goals (such as navigating to a particular destination or investigating planetary geology) in to a series of *subtasks* that serve as primitive operations [6]. Subtasks resulting from a navigation mission goal might include planning segments each implementing a portion of the total goal, while subtasks associated with planetary geology might include deploying and operating a sampling attachment. The *task executive* is responsible for carrying out each subtask; while the subtask is in progress, the *task monitor* [7–9] is observing the progress and assessing the potential for the executive to reach the subtask's end state within performance specifications. For an unreachable end state resulting in autonomy failure in a contemporary deep space robotic system, mission capability would be sacrificed in order to place the vehicle into safe mode and await further instructions [10]. Instead, in our model, the *assistant selector* is consulted to determine which of several potential *actors* (human assistants) might be called upon to rescue the executive. The rules by which an assistant selector chooses an assistant are called a *policy*. Ideally, the assistant's exemplar is used to both inform the assistant selection block via *machine learning* to improve the process of selection, while simultaneously informing the task executive about how to resolve the error condition in the future. In this work, we focus on the assistant selection and machine learning aspects, in red in Fig. 2.

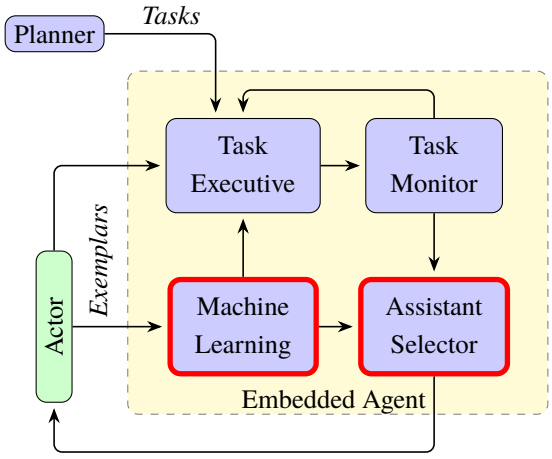


Fig. 2 Information flow in a model of an ideal learning robot. Our innovation is in the machine learning and assistant selection blocks, shown in red, used when subtasks cannot be completed.

Assistant selection can broadly be split into two categories: *static* policies and *dynamic* policies. Static policies designate a single human operator or functional role to supervise and assist robots at all times [11, 12]. These policies may be grounded in expert opinions and preliminary analysis, as in [13], to establish an acceptable baseline level of performance. Fundamentally, a static policy is not capable of adjusting to changing performance or contextual conditions of the available assistants.

Instead, dynamic assistant selection policies are potentially capable of exploiting IOOs about the assistant population, the task to be completed, as well as the operating environment. This responsiveness allows a dynamic selection policy to respond to instantaneous and long-term changes in operational conditions without encoding explicit constraints or

capabilities.

A. Formal Assistance Allocation Problem Statement

Assume there exists a set $\mathcal{A} = \{a_v\}_{v=1}^{N_A}$ of actors capable of assisting an embedded agent running onboard a robot during an autonomy failure. The embedded agent a_u maintains a history from timestep $k = 1$ to the present of past subtasks t_k that have failed, assistant allocation decisions, and resulting reward earned by each decision. At timestep $k = j$, embedded agent a_u must assign a single actor a_j from assistant set $\mathcal{A}$ of size N_A to recover from failed subtask t_j and maximize mission utility U, given only the partially observed history from timesteps $1, ..., (j-1)$ of past assignment decisions and resulting rewards. That is, maximize $U = \sum_{k=1}^{\infty} \gamma_k \mathbf{r}_k$, where γ_k with N_A elements each $\in \{0, 1\}$ is a vector of indicator values denoting the selection decision for each timestep k and $\mathbf{r}_k$ is the reward vector earned by every agent at timestep k, subject to the condition that only the selected element of $\mathbf{r}_k$ is observed at each step k. The rules for composing γ_k describe a policy. One potential solution is shown in Fig. 3 in the form of an *index policy*; this solution calculates a metric (expected reward in this case) and chooses an actor corresponding to the largest expected reward.

We impose several simplifications and restrictions to reduce the complexity of our problem. No subtask may only be assigned to more than one actor, nor are robots allowed to help one another. Further, while any subtask may fail, failure likelihoods are unknown a priori. Finally, rewards are unobserved for unassigned actors.

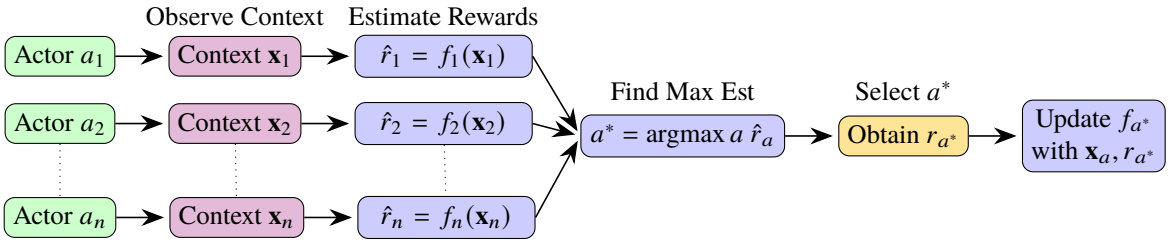


Fig. 3 A solution to the assistant selection problem in the form of an *index policy*. At each timestep, a single actor a^* with the highest estimated reward under the observed context is chosen to perform the task. After task performance, the actual reward r_{a^*} is used to update the actor's performance estimator.

B. Related Work

Dynamic assistant selection can be framed as an optimal allocation problem based on collective utility maximization for a single robot over the set of actors (i.e. possible assistants, where an actor's contribution to system utility is defined through a set of local utility functions based on each actor's state and assistance outcomes). Classical centralized [14, 15] and distributed [16] task allocation schemes require known utility functions to model the overall expected payoff of actor-task assignments. Importantly, most allocation schemes assume that the underlying task execution process and associated utilities are well-modeled and time-invariant, i.e. they are brittle and cannot respond to the system's actual performance [17].

Optimal assignment problems under uncertainty have also been addressed through partially observable Markov decision processes (POMDPs), which are typically solved offline to determine an online actor assignment policy. In alternative formulations, POMDPs have been used to optimize spatio-temporal assignments of robots to tasks [18] and other human-robot collaboration efforts [19]. POMDPs require accurate system models to evaluate both actions and rewards; however, such models are often unavailable or challenging to develop for long-term autonomous robotics scenarios. Optimal allocation policies are also extremely difficult to find for POMDPs with high-dimensional state spaces [20].

Reinforcement learning (RL) can adaptively integrate feedback from action choices and improve future choices. Parker, et al.'s behavior-based L-ALLIANCE [21] is an early example of an RL-based task allocation system. A key limitation of that work is that it considers time to complete objectives as the only performance criteria, when in fact it is only one of many factors that influence overall task and mission performance. Nevertheless, RL algorithms that balance exploration and exploitation allow contextual and performance data from assigned actors to be leveraged over time, without requiring accurate *a priori* knowledge of system models, making it attractive for missions involving model uncertainty.

Our previous work used a hybrid linear contextual multi-arm bandit [22], using *indicators of opportunity* as *context features* to inform assistant decisions. In our multi-arm bandit formulation of the assistant selection problem, each assistant is represented by an *arm*; the assistant selection problem is solved by *exploring* over the space of arms and contextual information and then *exploiting* the learned information to accumulate reward. The hybrid bandit formulation allows us to consider observations that affect all actors, as well as individual observations capturing unique human responses. We validated the utility of contextual information in making decisions by comparing a non-parametric multi-arm bandit approach (KLempUCB, [23]) to a context-aware approach, finding a marked performance improvement.

Zero-shot learning techniques have recently gained popularity as a method of bootstrapping neural networks from small datasets to estimate rewards and potentially be of use in assistant selection. One challenge to the use of zero-shot learning is the typical application to classification problems, where the 'output' is a probability of an input representing a particular class. The approach in [24], learns feature dependencies in a strikingly similar manner to that of linear contextual bandits. Both techniques learn the sensitivity of the predicted outputs to changes in the input feature space. However, the zero-shot learning approach doesn't include provisions for exploration, that is, directing current choices to improve future estimates.

From operations research literature, we find a parallel problem in the technician selection problem [25], where a set of technicians must be dispatched to a set of service calls while considering the experience and learning of the servicing technicians. One key distinction between the assistant selection problem and the technician selection problem is that the set of service calls is known *before* assignments must be made; in our problem formulation, we do not have prior information on the frequency or nature of the autonomy failures that will need to be addressed.

C. Complexities in Human Modeling: *Non-stationarity*

This work builds on our previous work by framing a multi-arm bandit reinforcement learning framework to model the *non-stationary* reward distributions associated with human learning after repeated experience. Non-stationarity refers to a property of reward distributions indicating that such distributions change in time. The fact that humans learn and improve their performance through repeated practice is non-controversial. However, the exact nature of the human learning process falls in the realm of cognitive psychology and does not readily lend itself to generative modeling. The seminal work in the field was proposed over one hundred and thirty years ago by Hermann Ebbinghaus [26] who proposed a theory of learning and forgetting; while his sole experimental subject was himself, the nature of human learning is still an active area of research. Other notable historical work (merely one hundred years old) is the Yerkes-Dodson effect [27] based on analyzing learning in mice. While more contemporary work has expanded on these original concepts [28], proposed models of human learning are still based on empirical post-hoc observation rather than generative processes grounded in biology. The typical application of these post-hoc models is in estimating production; without a generative process, determining appropriate parameters under which the model will be valid becomes problematic.

A more relevant example to aerospace applications can be found in the Federal Aviation Administration (FAA)'s Aviation Instructor's Handbook [29]; while this material is originally designed to facilitate pilot training, the instruction principles are portable to arbitrary domains where humans are expected to learn. The FAA supplements E.L. Thorndike's original laws of learning [30] to give instructors a concise guide to instruction. These principles can be used as a guide to the types of experiences that an assistant selection system might need to monitor. We briefly recap these six laws:

Law of Readiness: A learner must be mentally prepared to learn, free from distraction or external worry.

Law of Effect: Learning is enhanced when performance feedback is positive and constructive, rather than belittling or frustrating.

Law of Exercise: Performance is gained by repeated practice of a skill in real-life scenarios and lost by disuse.

Law of Primacy: A learner's first experiences have a much greater impact than later experiences; unlearning bad habits takes more effort than learning proper technique initially.

Law of Intensity: Exciting scenarios have a greater impact than routine, boring training.

Law of Recency: A learner's skills fade with time and require constant practice.

The mere presence of such complicated learning dynamics is sufficient to motivate research into modeling *non-stationarity* in human performance distributions. In particular, we are interested in capturing the effects of the Law of Exercise and the Law of Recency.

D. Models of Human Learning

In our previous work [3], our simulation models were not based on observations of realistic human behavior. In this work, we improve the fidelity of our simulations by implementing several types of learning models as presented in [31]. All of these models attempt to capture dynamic aspects of human behavior, namely the performance changes associated with learning due to repetition and forgetting due to disuse. These models are justified with many observational studies to support their general form using data gathered over a wide population of typically industrial settings. In this work, we are interested in using these empirically derived models as generative models of human behavior. Agents implementing these generative models can then be used to explore the effectiveness of various assistant selection techniques under human-like learning dynamics.

1. Power Law

Originally proposed by Wright after observing the manufacture of aircraft [32], the power law proposes an exponential learning model to describe the time/resources needed to produce the nth unit in terms of a baseline cost t_1 and a learning constant b, as shown in Eq. 1.

$$t_n = t_1 n^{-b} \tag{1}$$

The performance of an agent with power law learning dynamics is shown in Fig. 4; importantly, we present the power law dynamics as an asympototic reward function rather than as the original decaying exponential cost function to provide a consistent comparison with other models. This figure illustrates the performance of a single agent repeatedly selected, with two periods of non-selection from timesteps 31-40 and 170-180. With the power law agent, learning is simply delayed due to non-selection; there is no attempt to model the forgetting process. Once the agent resumes being selected at timesteps 41 and 181, the reward model continues its asymptotic growth. In our agent, we use representative values for the learning constant suggested by [33]. One particular feature of this model is that the first execution of this model yields zero reward, implying that the baseline knowledge possessed by the actor at the beginning of the simulation is also zero.

2. Stanford B Model

Proposed by [34] to account for experience or talent present at the start of a new task to be learned, the Stanford B model introduces a new parameter B:

$$t_n = t_1 (n + B)^{-b} \tag{2}$$

that functions to improve the initial performance of the agent. The performance of an agent with a Stanford B model of learning is shown in Fig. 5. This performance includes a period of non-selection from times 25-34, during which

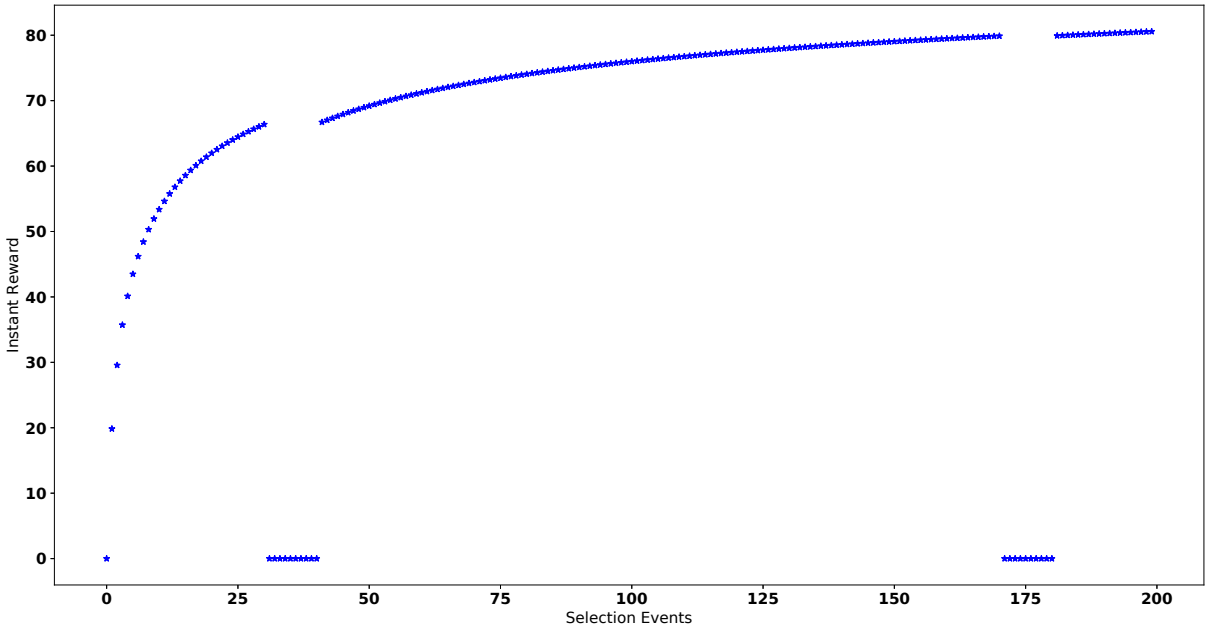


Fig. 4 Reward obtained through a power-law model as a function of selection events. Non-selection occurs at times 31-40 and 170-180.

the agent returns a zero reward and 'pauses' learning. The critical feature of the Stanford B model is observed at selection event 0; in contrast to the power law, the Stanford B model returns a parameterizable reward representing initial capability.

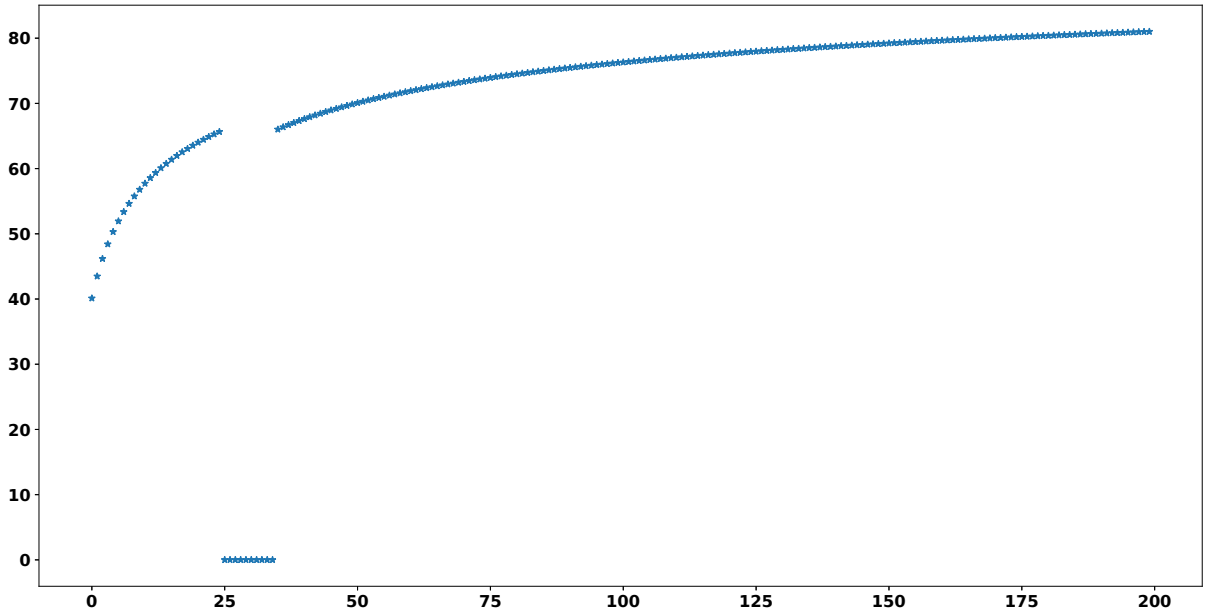


Fig. 5 Reward obtained through a Stanford B model as a function of selection events. Non-selection occurs at time 25.

3. Learning / Forgetting / Relearning / Fatigue

Neither the power model nor the Stanford B model account for the degraded performance associated with forgetting due to skill disuse. In our models, this disuse corresponds to a non-selection event. Once the agent is again selected, performance will return as necessary skills are re-learned. We use Dar-El's learning-forgetting-relearning model [31] as a foundation of a more complete model of the dynamics associated with both selection and non-selection. Representative behavior of a single agent with learning and forgetting events is shown in Fig. 6. In our implementation of Dar-El's model, we add in several features that are appropriate to application of this model to an individual's learning/forgetting cycle. Firstly, we include the base offset as provided in the Stanford B model, so that nonzero reward is earned at time zero to represent initial experience. Secondly, we include a fatigue term to reduce the rate of learning as the agent is repeatedly selected. In our agent, this fatigue term is reset immediately after a period of non-selection to simulate the impact of a rest period. Once the agent is selected once more, the learning level is estimated to have performance resume at an appropriate level and rate. Immediately after a forgetting episode, the returned reward is the lowest point on the forgetting portion of the performance curve (i.e. at times 55 and 90 in Fig. 6). In our formulation, this agent includes parameters for baseline performance, learning/forgetting/fatigue rates, initial experience, and an arbitrary context function.

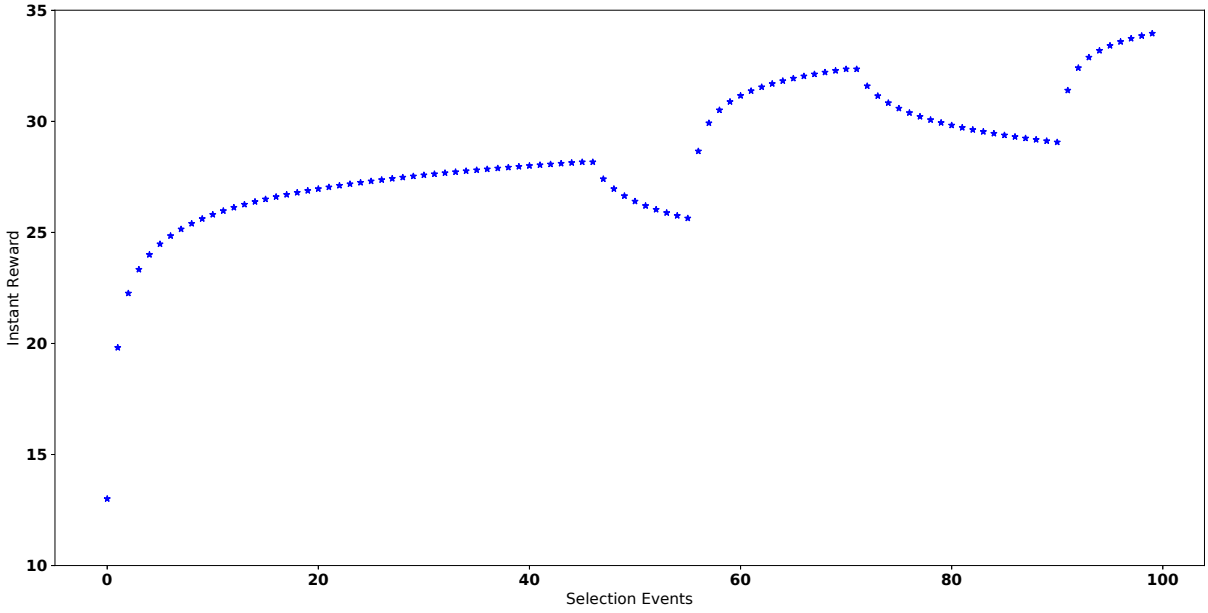


Fig. 6 Learning dynamics of the learning/forgetting/fatigue agent. Forgetting occurs at time 46-54 and 71-89. Learning and forgetting occur according to a power law.

III. Multi-Arm Bandits with Non-Stationary Models

In the vanilla multi-arm bandit formulation [22], a critical assumption is that reward distributions are unknown, but fixed. Several extensions of the multi-arm bandit problem have been proposed to study the impact of relaxing

this requirement of *stationarity*; briefly, the two types of non-stationarity are split into abrupt changes and gradual changes. Some examples of of methods proposed to model non-stationarities include Bayesian change detection in [35], a Gaussian mixture model with dynamically updated weights [36], and a hierarchical extension of [37]'s LinUCB [38].

As a solution form, we investigate one particular formulation based on Gaussian Process (GP)s[39]; this formulation has the benefit of being a natural extension of the fundamental Upper Confidence Bound (UCB) multi-arm bandit algorithm while being capable of incorporating contextual information. Within the framework of a general multi-arm bandit shown in Fig. 3, a GP serves the role of the per-actor reward estimation function f_i that accepts a context vector and produces an estimate of future reward. In our motivating application, context vector contents consist of task, environmental, and behavioral observations about each potential human assistant.

A. Contextual Gaussian Process (CGP) Multi-Arm Bandits

Gaussian processes naturally lend themselves to the contextual multi-arm bandit problem as a means of estimating an unknown reward function. Formulated as a multi-arm bandit, each arm may be its own process independent of others, or may be a shared process that models separate arms using an additional context variable. While the value of mutual information supported by the shared process formulation presents an interesting avenue of future research, we have used a per-agent independent formulation in this work for simplicity.

In the context-free case proposed by [40], a reward prediction at time t is obtained for a target agent a via an expression similar to that in the UCB vanilla multi-arm bandit case:

$$\hat{\mathbf{r}}_t = \underset{a}{\operatorname{argmax}}\, \mu_{t-1}(a) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(a) \tag{3}$$

with parameter β a system design choice to balance exploration versus exploitation. Note that in Eq. 3, the selection is among agents; the mean function and covariance function are operating over the space of available agents. In the extension proposed by [41], a context vector $\mathbf{z}_t$ is added to the GP's input, as:

$$\hat{\mathbf{r}}_t = \underset{a}{\operatorname{argmax}}\, \mu_{t-1}(a, \mathbf{z}_t) + \beta_t^{\frac{1}{2}} \sigma_{t-1}(a, \mathbf{z}_t) \tag{4}$$

In this work, independent GPs are used per agent, leading to the formulation:

$$\hat{\mathbf{r}}_t = \underset{a}{\operatorname{argmax}}\, \mu_{a,t-1}(\mathbf{z}_t) + \beta_{a,t}^{\frac{1}{2}} \sigma_{a,t-1}(\mathbf{z}_t) \tag{5}$$

This formulation reduces the dimensionality of the GP at the expense of removing the ability of one agent's experiences to potentially influence another agent's performance. Given the our simplifying assumptions, this limitation is not problematic, but presents a possible avenue of future development.

B. The Temporal Kernel

Gaussian processes as modeling tools are dependent on the choice of covariance kernel used to model the relationship between pairs of data points; we present a brief derivation of a particular type of kernel capable of modeling the difference in two measurements along a temporal axis. In general, the choice of kernel function is a critical decision by a system designer; indeed, choosing kernel functions in an automated manner is a subject of current research [42]. In [43], the idea of using Gaussian processes to model time varying processes, in which time participates as a dependent variable, was explored in datasets including temperature variation over time. In our derivation, we show the interpretability of this temporal kernel's hyperparameters as a special case of a standard kernel, the *exponential* kernel.

Stationarity as a property of a kernel type refers to whether the kernel is independent or dependent on the absolute values of input values $\mathbf{x}$. *Non-stationary* kernels are expressed as functions of their two inputs $\mathbf{x}$ and $\mathbf{x}'$, as $K(\mathbf{x}, \mathbf{x}')$. Stationary GP kernels rely on the *radius* $r = |\mathbf{x} - \mathbf{x}'|$ and are expressed as $K(r)$. We are interested in analyzing relative changes in performance and so confine ourselves to stationary kernels in the remainder of this work. In the scalar case, the radius is simply the absolute difference between the two data values:

$$r(x, x') = |x - x'| \tag{6}$$

In the multi-dimensional case where input dimensions are independent, the radius is defined with respect to input data x and x', with hyperparameters l_i as:

$$r(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\text{input\_dim}} \sqrt{\frac{(x_i - x_i')^2}{\ell_i^2}} \tag{7}$$

Should the inputs be dependent across dimensions, an additional hyperparameter Σ describing the covariance between dimensions may be introduced using distance measures such as the Mahalanobis distance:

$$r(\mathbf{x}, \mathbf{x}') = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_j)} \tag{8}$$

Given that the inputs appear in their differenced form r, stationary kernels are called Radial Basis Function (RBF) kernels and are expressed as functions of r, in the form $K(r)$. The exponential kernel, a standard kernel type, is defined with two hyperparameters: lengthscale l and variance σ, with (in the scalar case):

$$K(r) = \sigma^2 e^{\frac{-r}{l}} \tag{9}$$

The lengthscale hyperparameter l appears only in the scalar case; in the multidimensional case, this hyperparameter appears as l_i in Eq. 7, representing a per-dimensional scale, or as Σ in Eq. 8, representing a covariance between all

input dimensions. Note that the radius operation may be customized to the type of data under consideration and need not be limited to Euclidean distance.

The proposed temporal kernel is of the form:

$$K(t, t') = (1 - \epsilon)^{\frac{|t-t'|}{2}} \tag{10}$$

where the term ϵ models the impact of time as an exponential decay model, with $\epsilon \in [0, 1]$. Hyperparameter ϵ has an intuitive interpretation with respect to the time dependency between samples. For $\epsilon = 0$, samples are independent and uncorrelated between time steps; for $\epsilon = 1$, every time step yields a new reward function. Ideally, we can retain this intuition regarding hyperparameter effect on real-world performance using standard kernels.

Comparing the two forms of Eq. 10 and Eq. 9, we can express ϵ in terms of the exponential kernel's parameters. First. express the single-dimensional temporal kernel in terms of radius:

$$K(r) = (1 - \epsilon)^{\frac{r}{2}} \tag{11}$$

Equating Eq. 11 and Eq. 9:

$$(1 - \epsilon)^{\frac{r}{2}} = e^{\frac{-r}{l}} \tag{12}$$

Taking the natural log of both sides:

$$\frac{r}{2} \ln(1 - \epsilon) = \frac{-r}{l} \tag{13}$$

Cancelling a radius term:

$$\ln(1 - \epsilon) = \frac{-2}{l} \tag{14}$$

Finally, exponentiating and rearranging yields:

$$\epsilon = 1 - e^{\frac{-2}{l}} \tag{15}$$

Thus, we can use the standard exponential kernel and retain the interpretability of the hyperparameter ϵ with respect to correlation between time steps. With $\epsilon = 0$, the kernel is time-invariant; with $\epsilon = 1$, then each time step is independent of every other. In other work [44], we compare the performance of the temporal kernel to non-temporally aware GP estimates. Without including a temporal dimension in the GP's input context vector, the GP attempts to utilize all available data to estimate a mean and covariance function, leading to poor estimates that are not responsive to changing system dynamics. In contrast, including a temporal dimension allows the GP to track changes in underlying dynamics, preventing out-of-date observations from skewing the current predictions. All processes used in this work utilize a kernel consisting of an RBF kernel processing context values multiplied by an exponential kernel processing temporal

values.

IV. Gaussian Processes with Human-Like Dynamics

To investigate the ability of a temporally-enabled GP to model gradual non-stationary reward distribution dynamics such as those associated with human learning and forgetting, we utilize the learning/forgetting/relearning/fatigue model of Section II.D.3 with a single Gaussian process. In this section, we present a prediction surface plot sampled from the mean function of the GP using its current estimate, calculated only from its past observations. Our prediction surfaces interrogate the GP to provide reward estimates over a linearly sampled range of context values at each timestep. The prediction surface is a useful tool to visualize the quality of a GP's estimate, but is not directly used in our contextual multi-arm bandit framework. Estimates of reward are shown in a linear scheme increasing from purple to yellow. A single context variable is drawn from $\mathcal{N}(0, 1)$ at each time step for evaluation; these points are shown linearly fading from light blue to magenta to indicate older and newer samples respectively. The Gaussian process is re-optimized after each iteration whenever the process is simulating a human selection event using the value of the drawn context variable and the observed reward from the hidden reward function. During this re-optimization, internal hyperparameters are refined using maximum likelihood estimation to reflect the addition of this new data point. During iterations without a simulated selection event, no refinement occurs.

A. Context-Free Human-Like Dynamics

While our previous work emphasized the importance of the context vector in estimating actor performance, we begin with a simplified case in which the context function is simply zero.

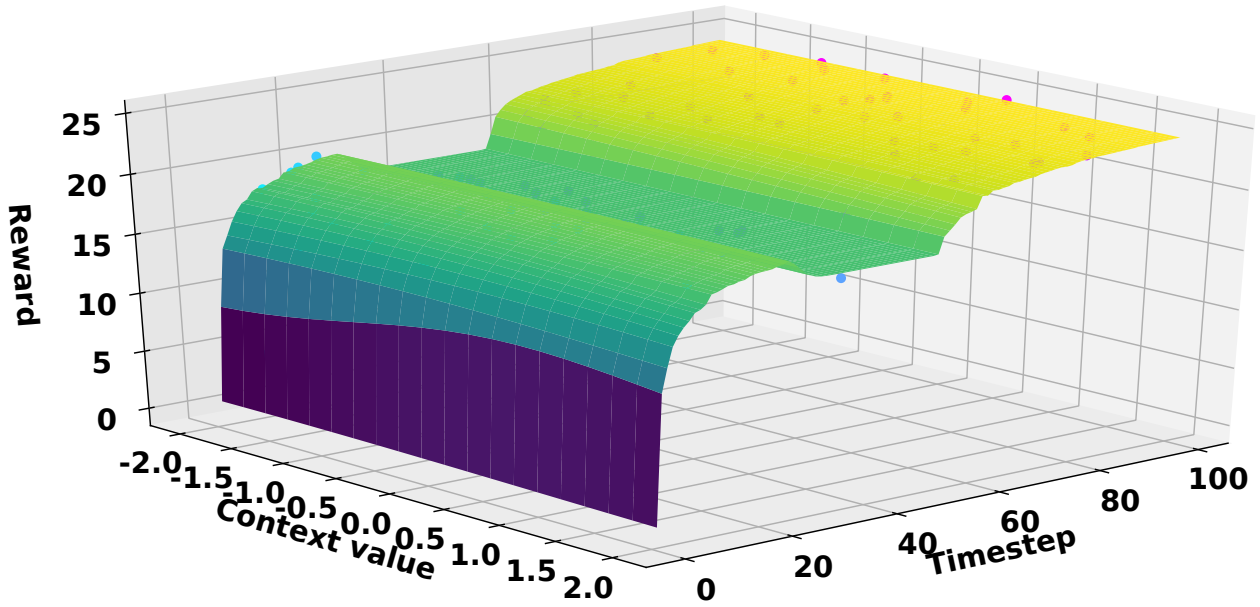


Fig. 7 GP estimating learning/forgetting/relearning/fatigue dynamics. Forgetting occurs at $t = 25$ through $t = 50$.

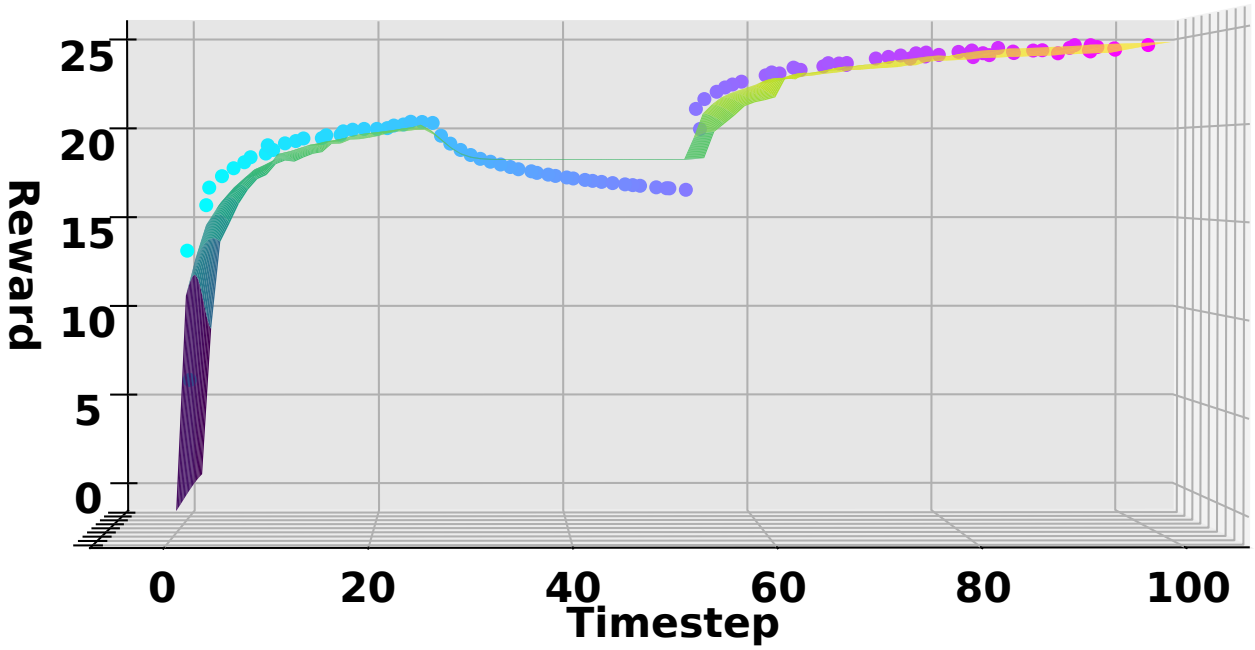


Fig. 8 GP estimating learning/forgetting/relearning/fatigue dynamics, side view, emphasizing the learning dynamics. Forgetting occurs at $t = 25$ through $t = 50$. The GP tracks the observed dynamics.

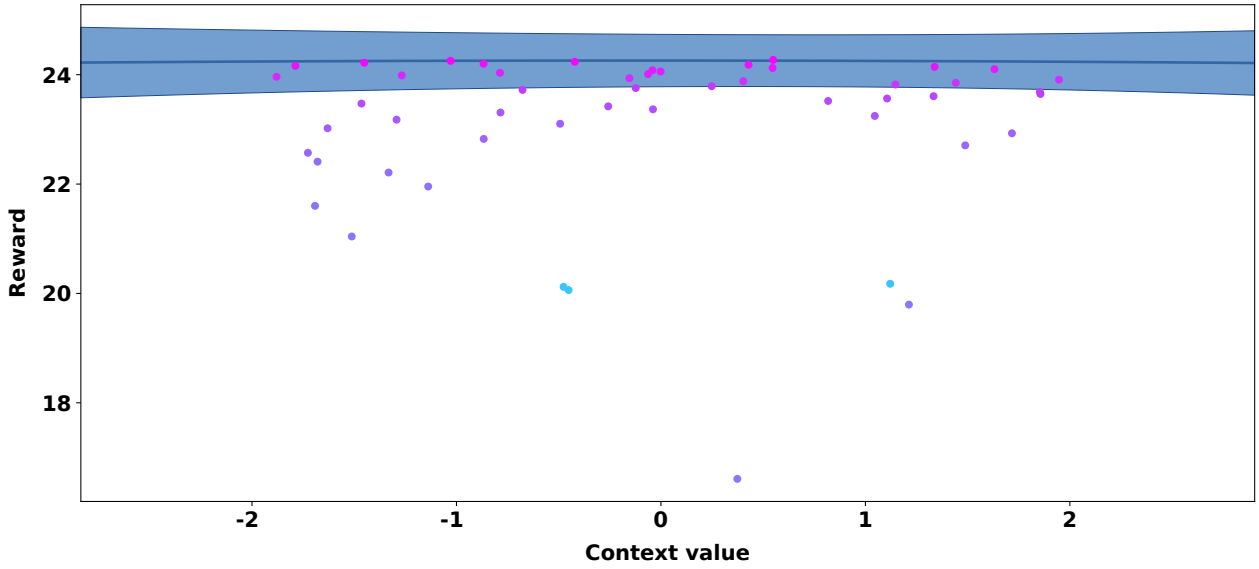


Fig. 9 Slice through the reward/context value axis at $t = 100$ emphasizing that past observations are downweighted compared to recent observations.

In Figs. 7 & 8, two views are presented of the prediction surface of the GP and a uniformly sampled context value over the interval $[-2, 2]$, from $t = 0$ to $t = 100$. At $t = 25$, a forgetting event is triggered such that new observations are no longer provided to the GP for inclusion into its internal model. However, the internal kernel is provided the current time step when a prediction is requested; this extra information is used by the GP to extrapolate future performance based on the temporal effects of outdated information. After approximately 10 time steps, the past information is so far outdated that the GP makes a constant estimate going forward, as indicated by the horizontal prediction surface in the

side view of Fig. 8. At $t = 50$, the notional forgetting process is ended, triggering the relearning phase. The GP is once again provided with observations and quickly resumes tracking the learning dynamics. At the conclusion of the simulation at $t = 100$, a slice through the context-value/reward axes in Fig. 9 reveals that the mean and one-σ variance estimates are based on the most recent data, downweighting past observations.

B. Contextual Human-Like Dynamics

Based on the importance of incorporating a dependency on contextual information into a reward estimate, we now introduce a context function to shape the agent's reward. Within the agent, the context function is evaluated and then summed with the learning dynamics. In these plots, we use a quadratic reward function with negative curvature.

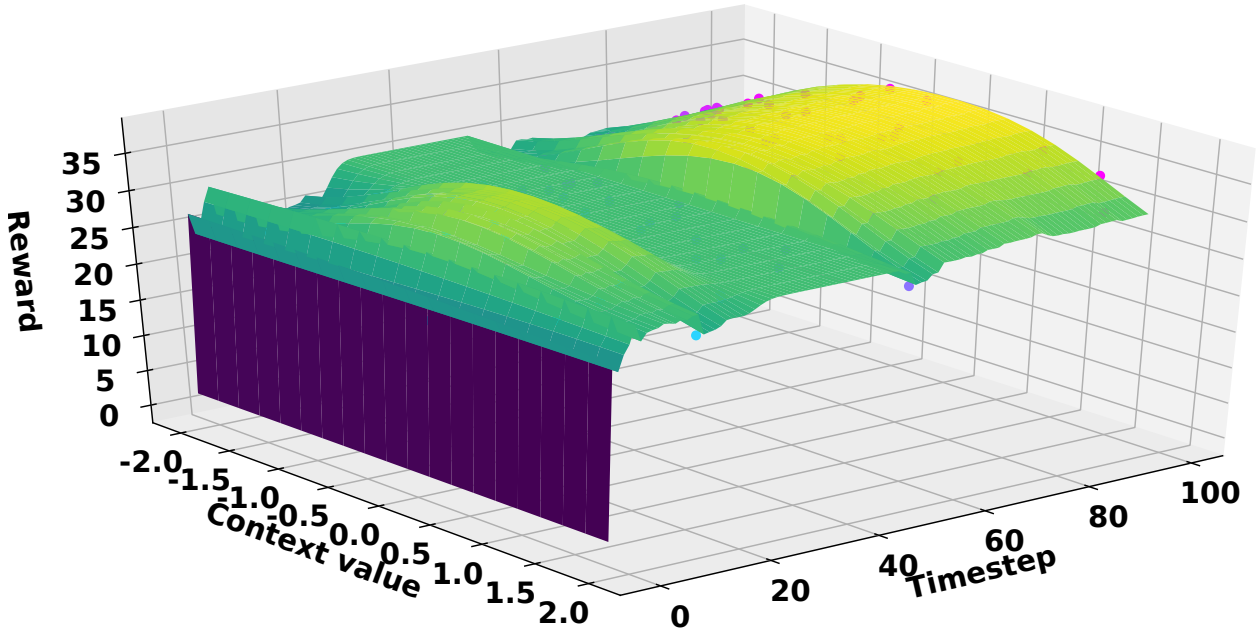


Fig. 10 GP estimating learning/forgetting/relearning/fatigue dynamics. Forgetting occurs at $t = 25$ through $t = 50$.

Figure 10 shows the prediction surface of the GP and a uniformly sampled context value over the interval $[-2, 2]$, from $t = 0$ to $t = 100$. At $t = 25$, a forgetting event is triggered such that new observations are no longer provided to the GP for inclusion into its internal model. The GP initially retains contextual dependencies, but downweights past information as forgetting occurs within the agent. Once agent selection resumes at $t = 50$, the GP recovers and resumes capturing the contextual dependency of the reward function. At $t = 100$, a slice through the context-value/reward axes in Fig. 11 again reveals that the mean and one-σ variance estimates are based on the most recent data.

V. Reference Selection Policies

To assess the usefulness of our CGP-based assistant selection algorithm, we require reference policies that are appropriate to a dynamic environment.

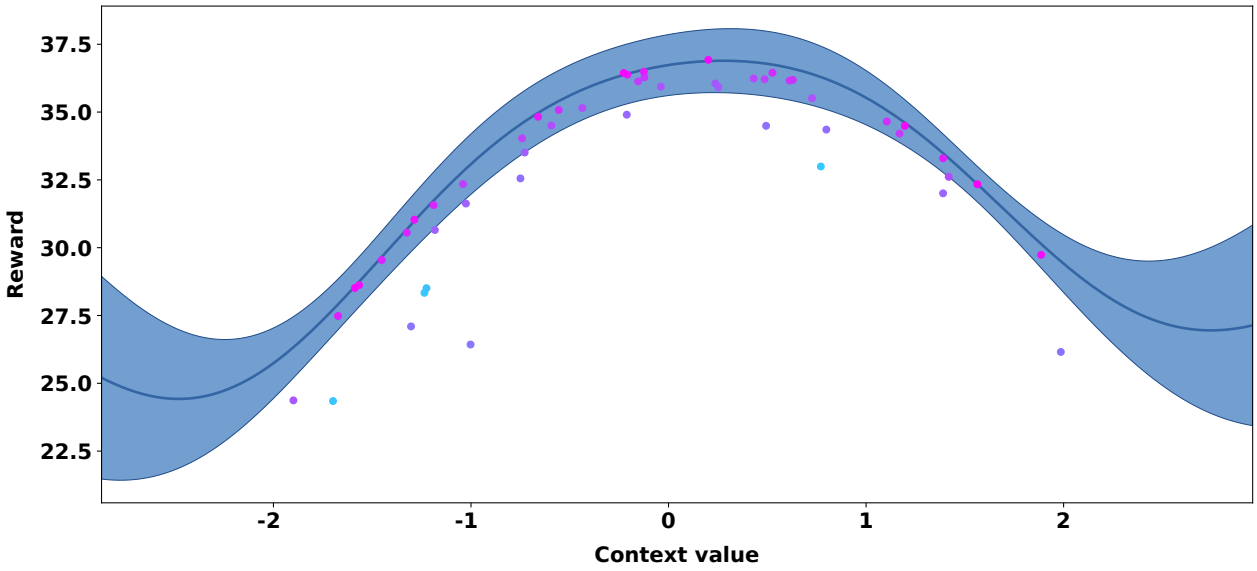


Fig. 11 Slice through the reward/context value axis at $t = 100$ emphasizing that past observations are downweighted compared to recent observations.

A. Standard Policies

The *random* policy uniformly selects from among all agents, ignoring any past selection history or context function dependency. The *solo* policy arbitrarily designates a single agent to be selected at every timestep, akin to the static policy of [13]. The *lockstep* policy selects each agent in turn, guaranteeing that every agent receives an equal number of selection opportunities at equal intervals. In addition to a standard ϵ-greedy approach, we also introduce a new type of policy that considers the recency of experience in addition to a performance estimate before selecting an agent.

B. ϵ-greedy With Contextual Gaussian Processes

In a standard ϵ-greedy selection policy, parameter ϵ represents a probability of exploring vs exploiting. In each round, a uniform draw is made to decide whether the bandit should explore potential rewards or exploit existing estimates of reward. In a system with dynamic (but unknown) reward distributions, exploring must occur throughout the selection horizon to ensure that reward distribution changes are captured and made available to the exploitation step. In our adaptation of ϵ-greedy, we use the contextual Gaussian process as the reward estimator while relying on external hyperparameter ϵ to choose between exploration and exploitation. If a decision is made to explore, our implementation selects an arm uniformly to pull. Should the decision be made to exploit, the arm with the largest potential reward according to the current Gaussian process estimate is pulled.

C. Alternative to Epsilon-Greedy: Proficiency Window

With the type of dynamics associated with human learning attached to our representative system, the standard ϵ-greedy action of uniform exploration is no longer appropriate, as there are dependencies associated with both forgetting

Algorithm 1 Proficiency Bandit Selection Algorithm

At each timestep t:
Neglect set $N \leftarrow \{\}$
$r \leftarrow \{\}$
for each arm i **do**
    Estimate reward r_i
    $r[i] \leftarrow r_i$
    if neglect time of $i >$ windowSize **then**
        $N \leftarrow N \,||\, (i)$
    end if
end for
if $N == \{\}$ **then**
    $a^* = \text{argmax}\, r$
else
    $a^* = \text{argmax}_{j \in N}\, r[j]$
end if
return a^*

and with dependencies on the context vector. As an alternative, we propose a *proficiency window* as a modification to standard ϵ-greedy approach. A proficiency window consists of a single parameter thresholding the time since last selection that bisects the set of all arms into two groups: those arms that are overdue for service and in danger of neglect, and those arms that are in some sense 'current' and ready to be exploited. Should the overdue set be non-empty, the standard index bandit function of argmax over estimated reward is used to determine which of the overdue arms should be pulled. Should the overdue set be empty, an argmax over estimated rewards is used, as in a standard index policy. Our alternative policy is outlined in Algorithm 1. The consideration of the overdue status of each arm differentiates our approach from that of a standard index policy, such as that described in Fig. 3.

This alternative policy addresses the main downside of the use of an ϵ-greedy exploration search in a contextual setting; we explicitly prevent neglect while at each time making the decision to gather the most estimated reward. However, the introduction of a new parameter, the size of the proficiency window, presents a challenge to the system designer. Should the window size be too large, arms will suffer performance estimation errors from excessive neglect; should the window be too small, the proficiency bandit will not have sufficient degrees of freedom from which to make informed choices. In the case of a window size smaller than the number of arms in the system, the proficiency window policy degenerates into the lockstep policy, albeit a lockstep that considers contextual information. In effect, the size of the window is a control parameter to express the relative importance of maintaining team proficiency (preventing neglect) and maximizing obtained reward.

VI. Experimental Validation In Simulation

In our previous work, we used the metric of *cumulative regret* to evaluate different policies in a multi-arm bandit experiment. For the learning and forgetting dynamics introduced in this work, cumulative regret is not a complete

description of the quality of the policy. Consider the case of the solo policy, where one actor is selected exclusively. According to cumulative regret, this policy is ideal, in that the policy selects the best possible outcome at each round. However, cumulative regret fails to capture the fact that the non-selected actors are neglected to the point of irrelevance. Instead, we propose the use of *cumulative reward* as a direct comparison of the ability of each policy to obtain reward, in concert with a distribution of the final *learning state* of each agent, as measured by selection events assigned to each agent. We are interested in a policy that maintains a minimum baseline of actor proficiency while also maintaining the ability to exploit opportunities to gain reward.

A. Setup

To demonstrate the effectiveness of the proficiency-aware contextual Gaussian process bandit, we conducted a series of Monte Carlo simulations exercising each of the policies in Section V. Our experiment investigates whether the proposed proficiency window policy obtains significantly more reward than competing policies. Each of the variations described below uses a fixed actor set size of 5 and finite time horizon of 100 selection events. In our simulation, we use selection events separated by uniform time intervals; this simplification is sufficient to exhibit the learning and forgetting dynamics of our simulated agents and demonstrate the viability of our technique without introducing the complexity of selection event distribution. As a followon, since our simulation agents are independent of one another, our simulation could be used to model real-world selection event progressions by including timesteps where no agents are selected.

Each instance of a simulation used separate parameterizations with normally-distributed values around nominal means within a learning/forgetting/relearning/fatigue model as described in Section II.D.3. Each model was further equipped with a unique context function to model a dependency on external variables; this context function is only available to each selection policy indirectly through reward observations.

Recognizing the importance of contextual information to potentially inform selection decisions, a sample context vector of length two was prepared in each round on a per-agent basis. The first entry was uniformly distributed over the space $[-2, 2]$, while the second entry was populated with the number of rounds since the last selection of that agent.

Each agent is selected repeatedly for an initialization period to provide a baseline set of training data for the Gaussian process to begin providing predictions.

For each simulation, a one-way analysis of variance (ANOVA) test on the cumulative reward at the given time step for each policy was conducted to determine if a particular policy yielded a significant difference in reward, followed by post-hoc comparisons if needed.

B. Baseline

In the baseline case, we compare the performance of the proficiency window policy to each of the alternative policies in Section V. The reward earned by each policy over 50 trials, with five actors, proficiency window of 4, and 100 events

per trial is shown in Fig. 12. Our ϵ-greedy approach used an ϵ of 0.2. By selection event 100, there is a statistically significant main effect on cumulative reward, F(4, 245) = 158.88, p < 0.001, η^2 = 0.72. Post-hoc comparisons using Tukey's HSD test revealed that the both the proficiency window policy and the ϵ-greedy policy were significantly better in terms of earning reward than every other policy (P < 0.001); ANOVA analyses are shown in Fig. 13.

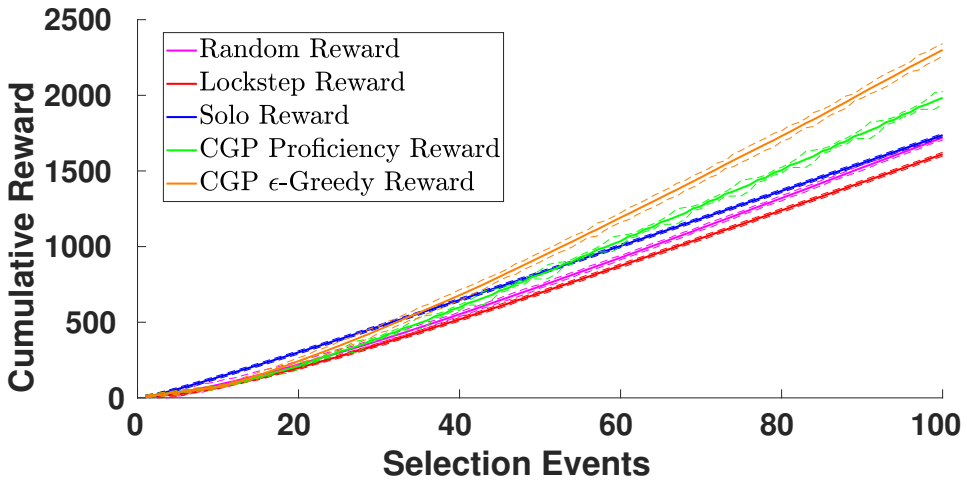


Fig. 12 Baseline performance of a proficiency-aware contextual Gaussian process multi-arm bandit with uniform selection event times. $\pm 5\sigma$ bounds are shown in dashed. By timestep 100, the proficiency window policy and ϵ-greedy policy have obtained significantly more reward than alternatives.

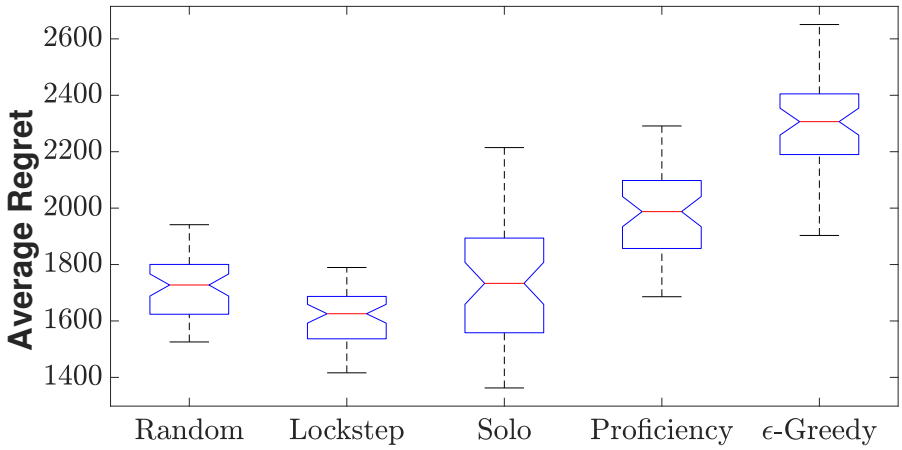
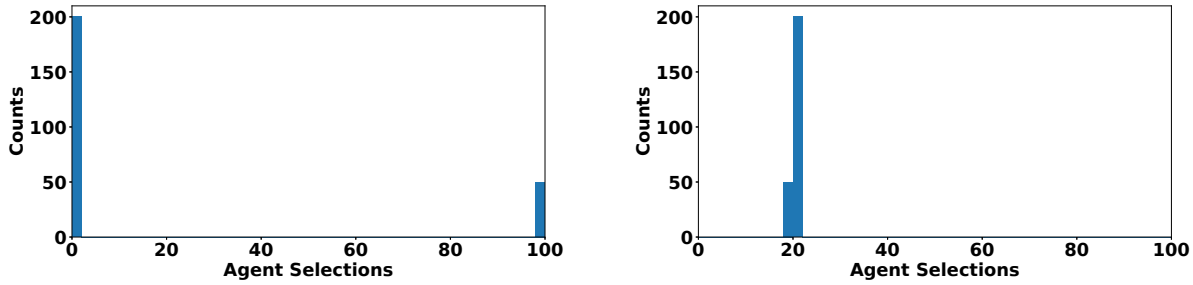


Fig. 13 ANOVA results at t = 100, indicating that the proficiency window policy and the ϵ-greedy policy have statistically significantly better results than the alternative policies.

To determine the relative agent neglect induced by choice of policy, we examined the distributions of learning opportunities in the form of selection counts as shown in Fig. 14 through Fig. 16. As opposed to the cumulative reward plots of Fig. 12, these results were compiled as a distribution of the n = 250 individual agent's selection counts at the end of their simulations, organized by policy. Under our agent model, selection counts are equivalent to learning opportunities considering each agent per trial to be independent. For example, in the *solo* policy's histogram (Fig. 14a), 50 agents were selected 100 times (corresponding to the single agent selected for the whole trial), while 200 agents were

selected zero times (corresponding to the neglected agents). The bi-modal distribution of the learning opportunities under the proficiency window policy (Fig. 16a) indicates that while a small number of agents are afforded the most selection events, no agent is completely neglected. While the solo policy of Fig. 14a is also bi-modal, the modes are at the extremes of zero selections and all selections, indicating that majority of the team was completely neglected. Comparing the proficiency window policy in Fig. 16a to the ϵ-greedy policy in Fig. 16b, the proficiency window policy places strict limits on the number of selections and prevents the partial neglect indicated by the lower mode of the ϵ-greedy policy.



(a) (b)

Fig. 14 Histograms of agent selection events by policy type: solo (Fig. 14a) and lockstep (Fig. 14b)

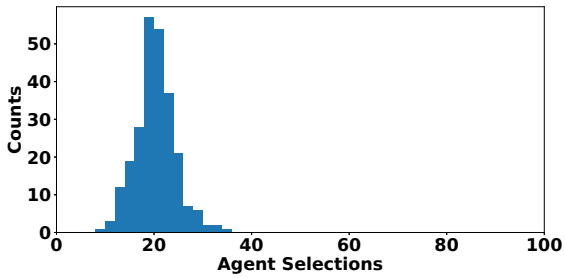
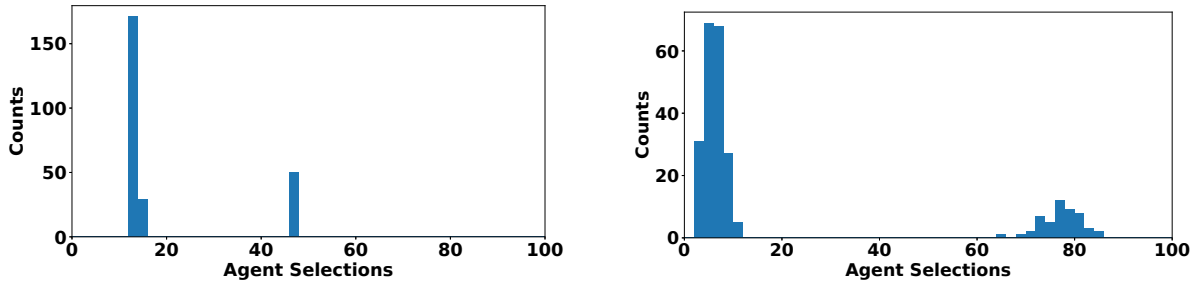


Fig. 15 Histograms of agent selection events, random policy.



(a) (b)

Fig. 16 Histograms of agent selection events by policy type: proficiency (Fig. 16a) and ϵ-greedy (Fig. 16b)

C. Effect of Window Size on Performance

Our next variation on the baseline case seeks to assess the impact of the size of the proficiency window on the performance of our approach. We express the proficiency window size as a function of the number of agents available (-2, -1, 0, +1, and +2) and compare performance to a baseline lockstep policy using three agents over 50 trials. As shown in Fig. 17, a window size too small relative to the number of agents yields inferior performance, as does an overly large window by permitting neglect to occur. We found that a window size change does have a significant effect on the cumulative reward compared to a lockstep baseline policy, $F(5, 294) = 4.56$, $p < 0.001$, $\eta^2 = 0.07$.

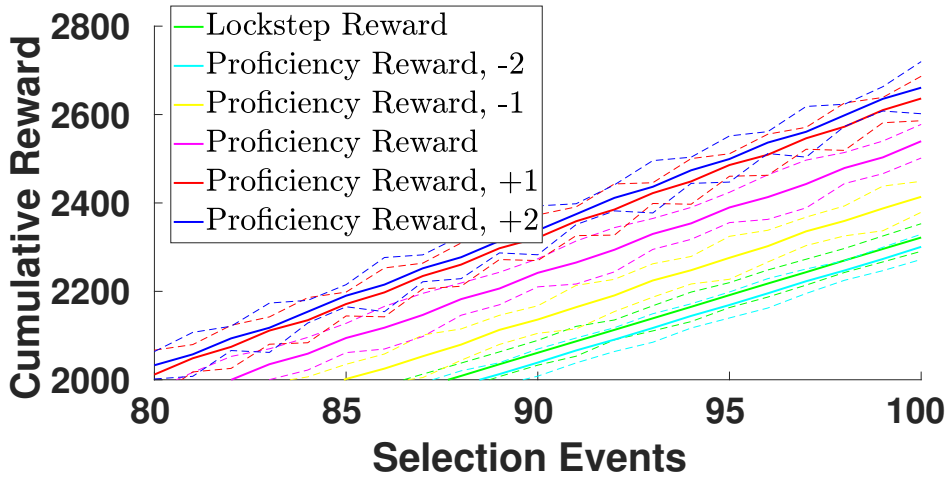


Fig. 17 Comparison of the choice of the window size with respect to the number of actors in the simulation. The +1 and +2 cases have a significantly improved performance (p=0.03 and p=0.01, respectively) compared to a lockstep policy.

VII. Discussion and Future Work

In the vanilla multi-arm bandit formulation, a core tenet is the idea of stationarity, such that non-selected arms retain their performance characteristics. Once human-like learning and forgetting dynamics are introduced, this core assumption becomes invalid. Instead, we must now consider neglect as a policy dependency. From a larger system perspective, the idea of maintaining a baseline level of proficiency in each human assistant while allowing superior performance to shine through whenever possible seems like a desirable goal; in practice, quantifying this concept requires new tools to implement the desired behavior. In particular, we must now consider the value of the team's experience as a whole, in contrast to the agent-myopic viewpoint typically used in standard multi-arm bandit formulations. This team-wide growth has been referred to as *plant learning* in operations literature [31], to describe the transfer learning among agents that occurs through informal means. From the operations perspective, learning is not being modeled individually, but rather on a wholesale basis among the entire cohort of agents. We posit that a hybrid approach might be required in such situations, where a reinforcement-learning approach includes both individual and team components in a reward function to be maximized.

Of noteworthy mention is the excellent performance of the ϵ-greedy policy with a contextual Gaussian process estimator. Indeed, based purely on our metric of cumulative reward, the apparent preferred choice should be the ϵ-greedy policy. While our proficiency window policy still yields an improvement over the alternative naive policies, a system designer seeking to maximize reward should choose the ϵ-greedy formulation. However, this performance comes at a price; there are no guarantees in an ϵ-greedy policy about maintaining relative team proficiency. While improbable, in principle an ϵ-greedy approach could degenerate into any of the naive policies, potentially leaving members of the team completely neglected. We interpret accepting slightly reduced performance of the proficiency window policy as a design tradeoff to guarantee overall team proficiency in return.

From a modeling perspective, the assistant selection system designer is faced with a choice regarding the flexibility desired of the underlying prediction model. While we used a well-known model of human learning in this work, empirical evidence in a variety of applications has shown that hyperparameters must be estimated in order for the model to be useful. While the development of a new kernel type for a GP incorporating a common model of learning dynamics may seem appealing, there is a design tradeoff between flexibility and usefulness. For example, in the sample plots of Fig. 8, during a forgetting phase the GP does not accurately model the agent's performance decay. In order to do so, it would require a kernel to capture the nature of human learning and thereby rigidly enforce a model that may not be accurate for individuals. Instead of the generic RBF-style kernels we have used to date, we could use a performance kernel capable of estimating the learning/forgetting dynamics using internal hyperparameters. Such a kernel would potentially reduce the recovery time required for an RBF-style kernel to respond to performance increases due to learning events, as well as more accurately model the performance decay due to non-selection. As future work, we would like to develop such a kernel and validate its usefulness in human subjects experiments.

VIII. Conclusion

This work presented an assessment of the potential for a contextual Gaussian process (CGP) to solve the assistant selection problem for robotic task fault recovery, under agent dynamics representative of human task learning and forgetting. Using a novel proficiency window policy, we have shown that a CGP can outperform several competing policies using the metric of cumulative reward, with the desirable side effect of ensuring that all members of a human-robot team are offered sufficient learning experiences to effectively contribute to team success.

While humans and robots exploring a planetary surface together may not be immediately possible, our work opens up the potential for broad impacts in the space robotics domain. For instance, instead of considering the autonomy of a planetary rover, we might consider the autonomy behind ground support tools used to generate the stored plans used to control daily operations. Such autonomous ground support tools face formidable challenges, from flight rule concerns to satisfying 'horse trading' between science and operations teams. Our work could be applied as a decision support tool to aid in conflict resolution, to identify the best person suited to help resolve conflicting goals. Alternatively, our

work might serve as the basis for a training management tool to ensure long-term mission continuity while adapting to dynamic personnel changes. One extension of our work that will be a required feature of such a tool is a means of 'warm starting' the reinforcement learning process to incorporate prior agent experience. While we considered an abstract simulation of a human-robot team, our foundational work is applicable to any scenario where a set of highly valued humans has a shared responsibility to oversee autonomous operations.

Funding Sources

S. McGuire was supported by NASA under grant NNX15AQ14H. C. Heckman and S. McGuire receive current support from DARPA under grant HR0011-18-2-0043.

References

[1] Mishkin, A. H., Limonadi, D., Laubach, S. L., and Bass, D. S., "Working the Martian night shift-the MER surface operations process," *IEEE Robotics & Automation Magazine*, Vol. 13, No. 2, 2006, pp. 46–53. https://doi.org/10.1109/MRA.2006.1638015.

[2] Rabideau, G., and Benowitz, E., "Prototyping an onboard scheduler for the Mars 2020 rover," 2017.

[3] McGuire, S., Furlong, P. M., Heckman, C., Julier, S., Szafir, D., and Ahmed, N., "Failure is not an option: Policy learning for adaptive recovery in space operations," *IEEE Robotics and Automation Letters*, Vol. 3, No. 3, 2018, pp. 1639–1646. https://doi.org/10.1109/LRA.2018.2801468.

[4] McGuire, S., Furlong, P. M., Fong, T., Heckman, C., Szafir, D., Julier, S. J., and Ahmed, N., "Everybody Needs Somebody Sometimes: Validation of Adaptive Recovery in Robotic Space Operations," *IEEE Robotics and Automation Letters*, Vol. 4, No. 2, 2019, pp. 1216–1223. https://doi.org/10.1109/LRA.2019.2894381.

[5] Drake, B. G., Hoffman, S. J., and Beaty, D. W., "Human Exploration of Mars, Design Reference Architecture 5.0," *Aerospace Conference, 2010 IEEE*, IEEE, 2010, pp. 1–24.

[6] Elfes, A., Weisbin, C. R., Hua, H., Smith, J. H., Mrozinski, J., and Shelton, K., "The HURON task allocation and scheduling system: Planning human and robot activities for lunar missions," *Automation Congress, 2008. WAC 2008. World*, IEEE, 2008, pp. 1–8.

[7] Sankaran, B., Pitzer, B., and Osentoski, S., "Failure recovery with shared autonomy," *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, IEEE, 2012, pp. 349–355. https://doi.org/10.1109/IROS.2012.6385524.

[8] Knepper, R. A., Tellex, S., Li, A., Roy, N., and Rus, D., "Recovering from failure by asking for help," *Autonomous Robots*, Vol. 39, No. 3, 2015, pp. 347–362. https://doi.org/10.1007/s10514-015-9460-1.

[9] Verma, V., Gordon, G., Simmons, R., and Thrun, S., "Real-time fault diagnosis [robot fault diagnosis]," *IEEE Robotics & Automation Magazine*, Vol. 11, No. 2, 2004, pp. 56–66. https://doi.org/10.1109/MRA.2004.1310942.

[10] Tipaldi, M., and Bruenjes, B., "Survey on fault detection, isolation, and recovery strategies in the space domain," *Journal of Aerospace Information Systems*, Vol. 12, No. 2, 2015, pp. 235–256. https://doi.org/10.2514/1.I010307.

[11] Fong, T., Thorpe, C., and Baur, C., "Robot, asker of questions," *Robotics and Autonomous systems*, Vol. 42, No. 3, 2003, pp. 235–243. https://doi.org/10.1016/S0921-8890(02)00378-0.

[12] Schreckenghost, D., Milam, T., and Fong, T., "Measuring performance in real time during remote human-robot operations with adjustable autonomy," *IEEE Intelligent Systems*, Vol. 25, No. 5, 2010, pp. 36–45. https://doi.org/10.1109/MIS.2010.126.

[13] Johnson, M., Bradshaw, J. M., Feltovich, P. J., Jonker, C. M., Van Riemsdijk, M. B., and Sierhuis, M., "Coactive design: Designing support for interdependence in joint activity," *Journal of Human-Robot Interaction, 3 (1), 2014*, 2014. https://doi.org/10.5898/JHRI.3.1.Johnson.

[14] Kuhn, H. W., "The Hungarian method for the assignment problem," *Naval research logistics quarterly*, Vol. 2, No. 1-2, 1955, pp. 83–97. https://doi.org/10.1002/nav.3800020109.

[15] Gerkey, B. P., and Mataric, M. J., "Sold!: Auction methods for multirobot coordination," *IEEE transactions on robotics and automation*, Vol. 18, No. 5, 2002, pp. 758–768. https://doi.org/10.1109/TRA.2002.803462.

[16] Bertsekas, D. P., "The auction algorithm for assignment and other network flow problems: A tutorial," *Interfaces*, Vol. 20, No. 4, 1990, pp. 133–149. https://doi.org/10.1287/inte.20.4.133.

[17] Zhu, D., Huang, H., and Yang, S. X., "Dynamic task assignment and path planning of multi-AUV system based on an improved self-organizing map and velocity synthesis method in three-dimensional underwater workspace," *IEEE Transactions on Cybernetics*, Vol. 43, No. 2, 2013, pp. 504–514.

[18] Bertuccelli, L. F., and How, J. P., "Active exploration in robust unmanned vehicle task assignment," *Journal of Aerospace Computing, Information, and Communication*, Vol. 8, No. 8, 2011, pp. 250–268. https://doi.org/10.2514/1.50671.

[19] Karami, A.-B., Jeanpierre, L., and Mouaddib, A.-I., "Partially observable Markov decision process for managing robot collaboration with human," *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*, IEEE, 2009, pp. 518–521. https://doi.org/10.1109/ICTAI.2009.61.

[20] Barry, J., Kaelbling, L. P., and Lozano-Pérez, T., "Hierarchical solution of large Markov decision processes," 2010.

[21] Parker, L. E., "Task-oriented multi-robot learning in behavior-based systems," *Intelligent Robots and Systems' 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference on*, Vol. 3, IEEE, 1996, pp. 1478–1487. https://doi.org/10.1109/IROS.1996.569009.

[22] Li, L., Chu, W., Langford, J., and Schapire, R. E., "A contextual-bandit approach to personalized news article recommendation," *Proceedings of the 19th international conference on World wide web*, ACM, 2010, pp. 661–670. https://doi.org/10.1145/1772690.1772758.

[23] Cappé, O., Garivier, A., Maillard, O.-A., Munos, R., Stoltz, G., et al., "Kullback–Leibler upper confidence bounds for optimal sequential allocation," *The Annals of Statistics*, Vol. 41, No. 3, 2013, pp. 1516–1541. https://doi.org/10.1214/13-AOS1119.

[24] Romera-Paredes, B., and Torr, P., "An embarrassingly simple approach to zero-shot learning," *International Conference on Machine Learning*, 2015, pp. 2152–2161. https://doi.org/10.1007/978-3-319-50077-5_2.

[25] Chen, X., Thomas, B. W., and Hewitt, M., "Multi-period technician scheduling with experience-based service times and stochastic customers," *Computers & Operations Research*, Vol. 82, 2017, pp. 1–14. https://doi.org/10.1016/j.cor.2016.12.026.

[26] Ebbinghaus, H., "Memory: A Contribution to Experimental Psychology," *Classics in the History of Psychology*, 1885. https://doi.org/10.1037/10011-000.

[27] Yerkes, R. M., and Dodson, J. D., "The relation of strength of stimulus to rapidity of habit-formation," *Journal of comparative neurology and psychology*, Vol. 18, No. 5, 1908, pp. 459–482. https://doi.org/10.1002/cne.920180503.

[28] Slamecka, N. J., "Ebbinghaus: Some associations." 1985. https://doi.org/10.1037/0278-7393.11.3.414.

[29] "FAA-H-8083-9A, Handbook, Aviation Instructor's," *US Department of Transportation Federal Aviation Administration*, 2008.

[30] Seligman, M. E., "On the generality of the laws of learning." *Psychological review*, Vol. 77, No. 5, 1970, p. 406. https://doi.org/10.1037/h0029790.

[31] Dar-El, E. M., *Human learning: From learning curves to learning organizations*, Vol. 29, Springer Science & Business Media, 2013.

[32] Wright, T. P., "Factors affecting the cost of airplanes," *Journal of the aeronautical sciences*, Vol. 3, No. 4, 1936, pp. 122–128. https://doi.org/10.2514/8.155.

[33] Carlson, J. G., and Rowe, A. J., "How much does forgetting cost," *Industrial Engineering*, Vol. 8, No. 9, 1976, pp. 40–47.

[34] Garg, A., and Milliman, P., "The aircraft progress curve modified for design changes," *Journal of Industrial Engineering*, Vol. 12, No. 1, 1961, pp. 23–27.

[35] Mellor, J., and Shapiro, J., "Thompson sampling in switching environments with Bayesian online change detection," *Artificial Intelligence and Statistics*, 2013, pp. 442–450.

[36] Yang, H., and Lu, Q., "Dynamic Contextual Multi Arm Bandits in Display Advertisement," *2016 IEEE 16th International Conference on Data Mining (ICDM)*, IEEE, 2016, pp. 1305–1310. https://doi.org/10.1109/ICDM.2016.0177.

[37] Lu, T., Pál, D., and Pál, M., "Contextual multi-armed bandits," *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 485–492.

[38] Wu, Q., Iyer, N., and Wang, H., "Learning contextual bandits in a non-stationary environment," *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, ACM, 2018, pp. 495–504. https://doi.org/10.1145/3209978.3210051.

[39] Rasmussen, C. E., "Gaussian processes in machine learning," *Summer School on Machine Learning*, Springer, 2003, pp. 63–71. https://doi.org/10.7551/mitpress/3206.001.0001.

[40] Srinivas, N., Krause, A., Kakade, S., and Seeger, M., "Gaussian process optimization in the bandit setting: no regret and experimental design," *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Omnipress, 2010, pp. 1015–1022.

[41] Krause, A., and Ong, C. S., "Contextual Gaussian process bandit optimization," *Advances in neural information processing systems*, 2011, pp. 2447–2455.

[42] Steinruecken, C., Smith, E., Janz, D., Lloyd, J., and Ghahramani, Z., "The automatic statistician," *Automated Machine Learning*, Springer, Cham, 2019, pp. 161–173. https://doi.org/10.1007/978-3-030-05318-5_9.

[43] Bogunovic, I., Scarlett, J., and Cevher, V., "Time-varying Gaussian process bandit optimization," *Artificial Intelligence and Statistics*, 2016, pp. 314–323.

[44] McGuire, S., "Autonomous On-line Learning of Assistant Selection Policies for Fault Recovery," Ph.D. thesis, University of Colorado at Boulder, 2019.