Fake It Till You Make It: Learning-Compatible Performance Support

Jonathan Bragg
Department of Computer Science
Stanford University
jbragg@cs.stanford.edu

Emma Brunskill
Department of Computer Science
Stanford University
ebrun@cs.stanford.edu

Abstract

A longstanding goal of artificial intelligence is to develop technologies that augment or assist humans. Current approaches to developing agents that can assist humans focus on adapting behavior of the assistant, and do not consider the potential for assistants to support human learning. We argue that in many cases, it is worthwhile to provide assistance in a manner that also promotes task learning or skill maintenance. We term such assistance Learning-Compatible Performance Support, and present the Stochastic Q Bumpers algorithm for greatly improving learning outcomes while still providing high levels of performance support. We demonstrate the effectiveness of our approach in multiple domains with simulated learners, including a complex flight control task.

1 INTRODUCTION

A longstanding goal of artificial intelligence (AI) is to develop agents can that augment or assist humans.[1] Using powerful tools like deep learning, researchers have recently made great progress toward building automated agents that can perform complex tasks at or above human ability. More limited but encouraging progress has also been made deploying these agents as assistants, for example, to help novice humans perform complex tasks like flying drones (Reddy, Dragan, and Levine 2018). In limited settings, researchers have even begun to consider higher-order effects such as human adaptation to the assistant, for the purpose of further improving the assistant's performance (Nikolaidis et al. 2017).

[1]While not the primary focus of this paper, one could assist machine agents instead, e.g., as in (Torrey and Taylor 2013).

While improving the ability of agents to assist humans performing tasks (i.e., to provide "performance support") is valuable, we argue that assistants should also be capable of *helping humans acquire and maintain skills relevant to the task*. In other words, in addition to providing performance support, we believe assistants should offer *learning* support. Education is a critical challenge in today's fast-changing world; such assistants could enable a new form of learning on the job, where the goal of the assistant is not only to ensure high-quality work, but also to support task learning by the human.

In this paper, we focus on the shared autonomy setting, which can enable high levels of initial performance. In shared autonomy, the assistant takes the human action as input and determines the final action (Reddy, Dragan, and Levine 2018). The assistant needs to consider the human action, since the human may have knowledge unavailable to the assistant (such as a goal location). In contrast to settings where the assistant provides advice only (Amir et al. 2016; Torrey and Taylor 2013), shared autonomy enables execution of complex tasks beyond the human's (initial) ability, such as flying drones (Reddy, Dragan, and Levine 2018). Our goal is to provide performance support that also facilitates learning, making the human less reliant on the assistant.

In this work, we formalize this goal as the design of Learning-Compatible Performance Support (LCPS). We first provide motivations for LCPS, and describe how learning assumptions about humans may influence outcomes. We then present Stochastic Q Bumpers, an algorithm for sharing control between a human and assistant, which significantly improves human learning (given a certain level of performance support), or team performance (given a desired learning level) for a natural class of learners—and does so by (1) considering projected rewards over full episodes and (2) executing assistant actions in an opportunistic, randomized fashion. Finally, we show the efficacy of Stochastic Q Bumpers compared to state-of-the-art shared autonomy methods (Reddy,

Dragan, and Levine 2018) across multiple domains, including a complex motor control task with deep reinforcement learning (RL) agents, and provide detailed behavior analysis for an environment with large negative rewards. We use simulated learners, which allow us to perform an extensive sensitivity analysis on the hyperparameters in the algorithms, in terms of their impact on performance and learning. We release our open-source code to facilitate future research.[2]

2 PRELIMINARIES & MOTIVATIONS

In this section, we briefly introduce Markov decision process notation and outline general motivations for designing assistants that provide Learning-Compatible Performance Support (LCPS), as well as special considerations for *transition* learners, the subclass of learners that we focus on in this work.

2.1 MARKOV DECISION PROCESS

We consider providing support for finite-length tasks occurring in episodic Markov decision processes (MDPs). An episodic MDP can be described by a set of states S, actions A, stochastic dynamics model $p(s' \mid s, a)$, reward model $r(s, a)$, and a discount factor $\gamma \in (0, 1)$. At least one state is a terminal state: transitioning to this state causes the process to reset to a (randomly selected) possible initial starting state. A decision policy π is a mapping from states to actions. The state-action value function of a policy π is denoted by $Q^\pi(s, a) = r(s, a) + \gamma \sum_{s'} p(s' \mid s, a) Q^\pi(s', \pi(s'))$, which represents the expected discounted sum of rewards of taking action a in state s and then following policy π. $V^\pi(s) = Q^\pi(s, \pi(s))$ and the optimal (highest value) policy, V, and Q are denoted as π^*,V^*, and Q^* respectively.

2.2 LEARNING-COMPATIBLE PERFORMANCE SUPPORT (LCPS)

If an agent is capable of providing performance support to the human, a natural question arises of whether it is even important to provide learning support at all (should performance support be "learning-compatible"?). We outline several affirmative reasons here.

Intrinsic benefits: People may derive personal value from being able to perform a task with decreased assistance. The benefits of human skill acquisition range from increased job opportunities and wages to personal pride and respect from peers. Even while receiving some performance support, humans may benefit from increased engagement and autonomy associated with learning.

Personalization: The assistant may be trained on a slightly different task or have undesirable biases from the perspective of the human. In this case, the human may wish to acquire proficiency based on the assistant's task definition, and then improve value alignment by operating independently of the assistant.

Sub-optimal assistants: The assistant may not be capable of performing the task with the desired proficiency. In this case, the human may first learn to perform the task at the level of the assistant, then exceed that level.

Meta learning: Learning to perform the task may help the human learn to perform other tasks more quickly.

Hierarchical learning: Learning to perform the task may help the human perform other tasks reliant on the current one.

Assistant failures: The assistant may fail, in which case it is desirable that a human be able to perform the task. Mechanical or software failures or limitations may cause the assistant to stop functioning or become unavailable. Worse, an adversary may take control of the assistant. Finally, the assistant may fail to generalize to new situations that the human is capable of handling (e.g., a self-driving car trained in the USA crosses over to Quebec, Canada and is unable to read signs written in French, while the human is bilingual).

Human deskilling: If the assistant takes responsibility for most actions, the human may lose performance ability, due to distraction, forgetting, or distributional shift. Unless the assistant can perform the task autonomously in all situations, safety could suffer.

Assistant costs: Assistants may be expensive. Software agents require computational resources, and robotic assistants may be costly to purchase or rent. Human learning that reduces dependency on the assistant should reduce these costs. Thus, LCPS is essential for ensuring equal access to possibly life-changing technology.

Privacy concerns: Using an assistant inherently requires sacrificing some privacy, since the assistant requires the details of the task in order to be helpful. Using the assistant to perform a smaller number of (possibly simulated) tasks may help to alleviate these concerns.

Task delays: Using an assistant can create delays, due to inherent processing time by the assistant or human, the communication time of the assistant or agent, or a combination thereof. For example, for language production, text may need to be produced by the human, communi-

[2] https://github.com/StanfordAI4HI/learning-compatible-performance-support

cated to the assistant, revised by the assistant, and processed by the human. Removing the assistant from the loop (facilitated by human learning gains) would significantly improve response times.

2.3 IMPORTANCE OF LEARNING & PERFORMANCE SUPPORT FOR TRANSITION LEARNERS

In a shared autonomy setting, the way that the human responds to agent control can have a significant impact on how and whether the human learns to perform the underlying task. Shared autonomy is an emerging area, and researchers do not yet have a clear understanding of how human learning works in this setting, and how the type of task may modulate learning. On the one hand, the standard learning-from-demonstration assumption, where learning occurs simply from observing the actions of another agent, may hold in some cases. For instance, a customer support representative interacting with a customer via a text interface may observe and learn from automated responses entered by a virtual assistant. On the other hand, educational research suggests that human learners may learn much more effectively if they execute actions on their own to internalize knowledge and acquire mastery (Koedinger et al. 2015), rather than simply observing the actions of others.

Humans may be unable even to make proper observations in shared autonomy settings. Consider, for example, shared autonomy for operating a complex system like a surgical robot arm or controlling audio production as a "disc jockey" (DJ). Operating these systems requires complex control of a rapid series of actions (controlling many motors or audio elements) in a "production" setting (e.g., surgery or a concert). It is likely that to maintain high performance, the agent would need to assert control frequently and without warning. In such settings, it would be difficult for the human to understand both when and how the agent may have modified the action they took, due to action complexity, action speed, and cognitive demands for the human participating in the task.

In this work, we focus on this model of learner, which Ho, Littman, and Austerweil (2017) term *transition learners*. Under this model, the human learns from experience as though all of his actions were executed, even though the assistant may have executed different ones. More formally, if the human executes action a_h^t in state s_t but the agent executes a different action a_t resulting in a transition to state s_{t+1} and reward r_t, the human will observe $(s_t, a_h^t, s_{t+1}, r_t)$, not the true (s_t, a_t, s_{t+1}, r_t). This model of learning is particularly challenging for providing learning and performance support, because learners can alias their wrong actions with the correct (agent executed) actions, which can harm learning. We believe that the transition model of learning is an important and realistic setting, that may become increasingly common as shared autonomy becomes feasible for many complex tasks. The flight control task we selected for our experiments is one example of a complex, fast-paced task where transition learning is likely to occur. Ultimately, brain control interfaces will likely enable incredibly complex, fast-paced control, where distinguishing between the human specified action and the actions taken (such as in the supported autonomy setting described by Downey et al. (2016)) will be very challenging.

3 STOCHASTIC Q BUMPERS ALGORITHM

The central challenge of our work is how to provide targeted performance support to transition learners in a way that enables them to achieve good performance, while also still enabling learning. Since under the transition model of learning, the human may assign credit to actions taken by the human when the assistant has actually executed a different action, the assistant must be strategic about when and how it provides performance support if it is to avoid the human learning a poor decision policy. Suppose, for example, that the assistant overrides any action taken by the human that is not optimal, in order to provide a high level of performance support. At best, the human will be unable to discover optimal actions since all actions result in optimal transitions. At worst, the human may learn that specific actions that lead to bad outcomes are very good, and learn a policy that—if executed without the assistant's corrections—could be significantly worse than random. Large amounts of experience executing such a policy may be very difficult to correct in the future, as it would require "unlearning" reinforced behaviors.

This example illustrates the problems of applying standard interactive strategies for learning with an expert (the assistant) in the loop. Those approaches can provably speed learning (compared to standard reinforcement learning) by first giving most control to the expert and subsequently giving the learner increasing amounts of control as it obtains better predictions based on the expert policy (Sun et al. 2017). This standard approach is also attractive from a performance support perspective, as it hopefully would transition to giving control to the learner only once the learner has learned a policy that is closer to the expert's. Unfortunately, this approach will not work for transition learners, as large amounts of expert control at the start will be difficult to overcome later.

An alternative approach that is likely to work much bet-

ter for transition learners is to enable the learner to take as many actions on their own as possible, but prevent large mistakes. Reddy, Dragan, and Levine (2018) provide one mechanism for sharing control with the human, by executing human actions whenever the agent's Q value (reward-to-go) for that action is "close" to the optimal reward-to-go; otherwise, the agent executes a near-optimal action. Our approach derives from two additional insights. First, instead of considering only reward-to-go, we seek to provide a minimum level of performance across the entire episode. Second, we provide a mechanism to avoid overriding the human many times in similar locations, with the goal of enabling the human to generalize better from a large, diverse sample of observations of the effects of executing their own actions.

We combine these insights to create the Stochastic Q Bumpers algorithm (Algorithm 1). To provide a minimum level of performance support, the algorithm takes as input a lower bound on possible episode rewards v_{min} and a parameter $\alpha \in (0, 1]$ that controls the level of performance support (smaller values give less control to the human). It then seeks to prevent episode rewards from falling below $V^*(s_0) - \alpha(V^*(s_0) - v_{min})$. As a measure of how close a human's action would bring the expected return to this lower value, the agent estimates

$$\hat{G}(a_t^h) = \sum_{i=0}^{t-1} \gamma^{i-1} r_i + \gamma^t Q^*(s_t, a_t^h),$$

the sum of rewards plus the expected reward-to-go of following the agent's policy π^*, where γ is a discount factor that enables fair comparison of $V(s_0)$ and future rewards. Now, instead of simply preventing any human actions where $\hat{G}(a_t^h) < V^*(s_0) - \alpha(V^*(s_0) - v_{min})$, the agent overrides with probability based on the distance to this lower value, where actions that completely eliminate this distance are overridden with probability 1 and actions that remove very little of this distance are overridden with probability close to 0. This stochastic overriding pro-actively injects high-reward actions intended to delay or prevent the agent from reaching a situation where most actions would fall below this lower value and the human would have very little control for the remainder of the episode. Further, the hope is that by stochastically overriding, the human will rarely take actions that are always overridden at particular states, and overrides will be distributed across a more diverse set of states that should interfere less with agent learning.

Algorithm 1 supports several different behaviors of overriding when the agent decides to do so. By default, $\text{AgentAction}(s_t) = \arg\max_a Q^*(s_t, a)$, where the $\arg\max$ operator selects a random best action. However, since overriding with the optimal action may make it difficult for transition learners to learn to execute the best action on their own, overriding with the second-best action may improve learning and provide a sufficient level of performance support. In this case, denoted $\text{AgentAction}(s_t, \text{SECOND})$, we impose a constraint on a s.t. $Q^*(s_t, a) < V^*(s_t)$, unless all actions have equal values (in which case it simply returns a random action).

To provide the human with increased control while $\hat{G}$ is still close to $V^*(s_0)$ and to increase the probability of overriding when $\hat{G}$ approaches the lower value (so that $\hat{G}$ is less likely to fall below this value), we apply a logistic transform f to the probabilities p of allowing the human action, as

$$f(p) = 1/(1 + \exp(k \cdot (p - 0.5))),$$

where $k = 2\log(1 - p_{max}) - \log(p_{max})$, and p_{max} is the desired probability for $f(1) = p_{max}$. This choice of k determines the steepness of the logistic function on the interval $p \in [0, 1]$, and transforms $p = 0$ to $(1 - p_{max})$

Algorithm 1 The Stochastic Q Bumpers Algorithm

$\alpha \in (0, 1] \leftarrow$ Parameter specifying amount of performance support
$Q^* \leftarrow$ Q values of the support policy
$f \leftarrow$ logistic probability transform (described in text)
$v_{min} \leftarrow$ lower bound on episode reward
$\gamma \leftarrow$ discount factor of Q values used in training
for $t = 0 : T$ (where T is the end of the episode) **do**
  $a_t^h \leftarrow$ action selected by the human
  $\hat{G}_t \leftarrow \sum_{i=0}^{t-1} \gamma^{i-1} r_i + \gamma^t Q^*(s_t, a_t^h)$
  Sample $p \sim [0, 1]$
  if $Q^*(s_t, a_t^h) \geq Q^*(s_t, \text{AgentAction}(s_t))$ **then**
    $a_t \leftarrow a_t^h$
  else if $p < f((V^*(s_0) - \hat{G}_t)/(\alpha(V^*(s_0) - v_{min})))$ **then**
    $a_t \leftarrow \text{AgentAction}(s_t)$
  else
    $a_t \leftarrow a_t^h$
  end if
  Execute action a_t
end for

function $\text{AgentAction}(s_t, \text{SECOND} = \text{false})$
  if $\text{SECOND} = \text{false}$ **then**
    return $\arg\max_a Q^*(s_t, a)$
  else if $\arg\min_a Q^*(s_t, a) = V^*(s_t, a)$ **then**
    return random action
  else
    return $\arg\max_a Q^*(s_t, a)$ s.t. $Q^*(s_t, a) < V^*(s_t)$
  end if
end function

and $p = 1$ to p_{max} ($p = 0.5$ remains unchanged).

4 EXPERIMENTS

Our experiments consist of analysis of behavior in a simple grid world setting, followed by empirical experiments in that domain and a complex motor control domain. We conduct all experiments by simulating human learners using reinforcement learning agents, and release our code for use by other researchers.[3] While we ultimately plan to conduct additional experiments with humans, simulated learners have several advantages for the current work from a controlled experiment standpoint. First, we can ensure that the learners always adhere to the learning assumptions of transition learners for the tasks we have selected. Second, we can obtain higher statistical power and reduce variance across learners by using the same learning algorithm and function approximation architecture for each learner. Third, simulation enabled us to do an extensive *sensitivity analysis*: by using simulated learners, we could scale experiments to the large numbers of learners required to sweep hyperparameter settings (Figures 2 and 4). This allowed the crucial comparison of how these parameter choices impact performance and learning. Finally, assisting machine agents is itself a useful end goal (Amir et al. 2016; Torrey and Taylor 2013).

In order to find strong baselines to compare with Stochastic Q Bumpers, we consider methods for providing performance support in shared autonomy that also enable various levels of human autonomy. As we have described, transition learners require autonomy to learn, as they need to see the results of their own actions in order to assign proper credit to actions. While previous work on shared autonomy shares control with the human for the purpose of improving performance rather than learning, we would expect such control sharing methods to improve human learning for transition learners, while also providing high levels of performance support. Reddy, Dragan, and Levine (2018) provide a simple, general method for sharing control, which we mentioned in the previous section and will call Local Q-Thresholding. Local Q-Thresholding overrides the human's action a_h^t whenever

$$Q'(s_t, a_t^h) < (1 - \alpha)V'(s_t),$$

where $Q'(s_t, a_t) = Q^*(s_t, a_t) - \min_a Q^*(s_t, a)$ and $V'(s_t) = \max_a Q'(s_t, a)$.

We also compare to the natural extremes, where the agent provides no support (which we expect to be good for eventual learning, but bad for initial performance) and where the agent provides maximal support (which we expect to be good for performance but bad for learning). Conveniently, these baselines correspond to Local Q-Thresholding with $\alpha = 1$ and $\alpha = 0$, respectively. Note that $\alpha = 0$ corresponds to maximal agent control only for $\mathrm{AgentAction}$ with $\mathrm{SECOND} = \mathrm{false}$.

[3] https://github.com/StanfordAI4HI/learning-compatible-performance-support

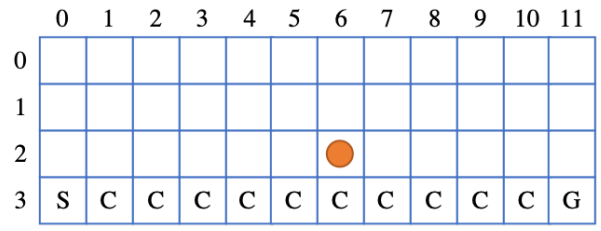


Figure 1: The Cliff Walking environment. S, C, and G denote the starting state, cliff, and goal, respectively. The orange dot denotes the agent / human's current location.

For clarity of our experiments, which are focused on the tradeoff between performance and learning support, we focus our evaluation on settings where the agent has already learned a good policy for the task. thus, we do not augment s_t with the human action a_h^t, as was done in (Reddy, Dragan, and Levine 2018). Further, when the agent takes control, we assume the agent executes the optimal action $\pi^*(s_t)$ rather than an action that is "closest" to the human's action, as our primary goal is to improve learning and not engagement.

4.1 CONTROL SHARING TRADEOFFS WITH LARGE NEGATIVE REWARDS

In order to better understand the differences between Stochastic Q Bumpers and Local Q-Thresholding, we first study the Cliff Walking grid world domain (Sutton and Barto 2018, p. 132). In this domain, every action (*left*, *right*, *up*, and *down*) incurs a reward of -1, except for moving off the cliff (e.g., by pressing *down* at the agent's location in Figure 1), which incurs a reward of -100 and resets the current location to the starting state. We add an additional $+100$ bonus for reaching the goal. Attempting to move off the grid results in -1 reward and no movement. The maximum episode length is 100.

We observe the choice of α for Local Q-Thresholding necessarily involves trading off between the level of support provided in the presence of very bad actions and the level of support provided when actions are relatively similar. For example, at the location of the agent / human in Figure 1 (2, 6), the Q values of the optimal policy are -13, 94, 92, 92 for *down*, *right*, *up*, *left* actions, respectively. In order to prevent the human from mov-

ing away from the goal (*up* or *left*), one would need to set $\alpha \leq (94-92)/(94-(-13)) \approx .019$ for Local Q-Thresholding. One position up, at (1, 6), the optimal Q values are 93, 93, 91, 91 for the same actions, respectively. Here, any value of $\alpha < 1$ will result in always overriding those actions. In fact, any value of $\alpha < 0.5$ will not allow the human to take any suboptimal actions locations that are not alongside the cliff.

We can see that two incompatible choices are possible. Setting α very close to 0 will prevent the human from moving backwards along the cliff toward the start state, but will not allow the human to take any other paths or sub-optimal actions away from the cliff, even if they cause very little harm and help learning. On the other hand, setting $\alpha > 0.5$ will allow the human to have better learning but can harm performance by letting the human undo all the progress they have made toward the goal.

Our Stochastic Q Bumpers method does not suffer from the same weaknesses, since it does not relying on local thresholding. Instead, it overrides when it is important to do so in the context of the suboptimality of the entire trajectory. If overall, the human has not taken many bad actions, they have more freedom (e.g., to move left along the cliff). If instead they have already taken many sub-optimal actions, they are prevented from moving too far from the goal.

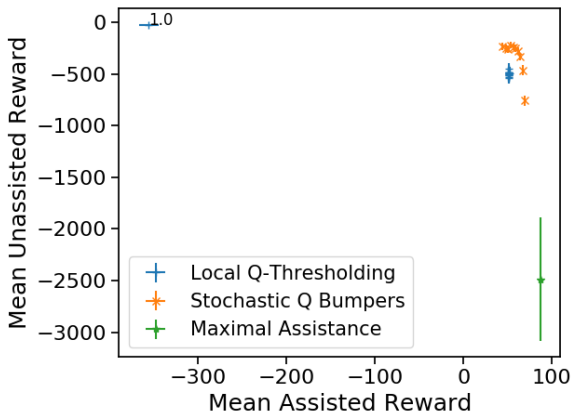


Figure 2: Average performance (unassisted vs. assisted) for the Cliff Walking domain. Fixing the assisted performance at the values obtained in the cluster of Local Q-Thresholding values (between 51 and 52 on the x-axis) and comparing to the best learning (unassisted) performance (top left, with no assistance), Stochastic Q Bumpers has 51% lower average regret than the best Local Q-Thresholding value. Similarly, when fixing learning (unassisted) performance and measuring assisted performance comparing to maximal assistance, Stochastic Q Bumpers has 43% lower regret than the best Local Q-Thresholding method.

4.2 CLIFF WALKING DOMAIN

We now present empirical results for the Cliff Walking domain described in Section 4.1. We conducted experiments using the OpenAI Gym (Brockman et al. 2016) environment interface, which we modified to include $+100$ reward at the goal. In this domain, we found that the second-best version of $\mathrm{AgentAction}$ works best, so we report results using that method for both Stochastic Q Bumpers and Local Q-Thresholding. For Local Q-Thresholding, we ran all values of α that result in different overriding behavior ($\alpha \in \{1, 0.5, 0.1, 0.0197, 0.0195, 0.0193, 0.0191, 0.0189, 0.0187, 0.0186, 0.0184, 0.0182, 0.0181, 0\}$). For Stochastic Q Bumpers, we ran $\alpha \in (0, 1.0]$ in 0.1 increments. Here and in all experiments, we set $v_{min} = 0$ for the lower bound on returns and $p_{max} = 0.999$ to determine the steepness of the logistic probability transform.[4]

We report mean results from 100 trials of 100 episodes each, using different random seeds for each trial. All trials used a tabular Q learner, with an epsilon-greedy exploration policy, where ϵ decays linearly from 1 to 0.02 over the first 10% of episodes, then remains at 0.02 (following the same DQN settings used in (Reddy, Dragan, and Levine 2018)). The learning rate decays according to the same schedule. We provided all assistant agents with the same optimal Q^* values, obtained using our tabular Q learning code run over a larger number of episodes. After each episode, we ran the learner with no exploration ($\epsilon = 0$) to measure its unassisted (learned) performance.

Figure 2 shows the mean unassisted performance vs. assisted performance (averaged over the entire trial), or learning support as a function of performance support. The plot shows the macro-averages across trials, with 95% confidence intervals based on standard error of the mean. In the top left, Local Q-Thresholding with $\alpha = 1.0$ corresponds to no assistance at all. As expected, assisted performance suffers but the learning performance is quite good. For comparison, we also include a point corresponding to always overriding sub-optimal values, regardless of whether the override value is second best (indicated in the bottom right of the figure as Maximal Assistance). As expected, learning suffers with maximal assistance (but assisted performance is very good).

Stochastic Q Bumpers strongly dominates Local Q-Thresholding in the top right. All Local Q-Thresholding points except $\alpha = 1.0$ have a value between 51 and 52 in terms of performance support (x-axis). However,

[4]We did not tune the value of p_{max} in our experiments.

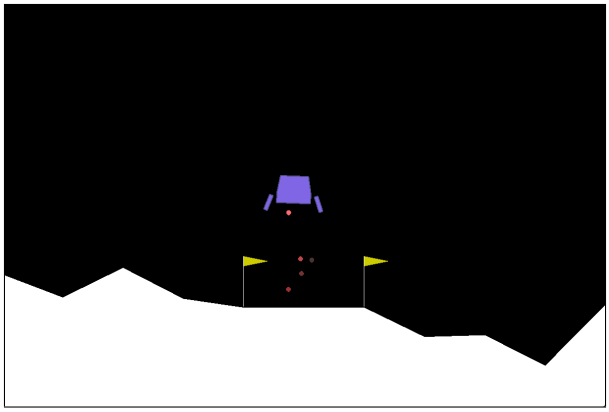


Figure 3: The Lunar Lander domain. Here, a Stochastic Q Bumpers ($\alpha = 0.2$) agent controlling the lander (the purple object with two protruding landing legs) is just beginning to override most actions of a random human learner by firing the main thruster (which emits red particles) to attempt to land successfully at the goal (between the yellow flags) and avoid a crash.

Stochastic Q Bumpers has a point ($\alpha = 0.7$) that is very similar in terms of performance support (53.9) yet obtains significantly better learning performance (-222.6 vs. -486.5 for the best Local Q-Thresholding value of $\alpha = 0.0195$.) Compared to the best learning performance value of -26.6 (Local Q-Thresholding with $\alpha = 1.0$), this is a reduction in regret of 51%. Similarly, if we fix a measure of learning performance on the y-axis, say -465.6 for Stochastic Q Bumpers ($\alpha = 0.2$) and -486.9 for Local Q-Thresholding ($\alpha = 0.0191$), Stochastic Q Bumpers has assisted performance of 66.6 compared to 51.3 for Local Q-Thresholding. Compared to the best assisted performance value of 87, this reduces regret by 43%.

4.3 LUNAR LANDER DOMAIN

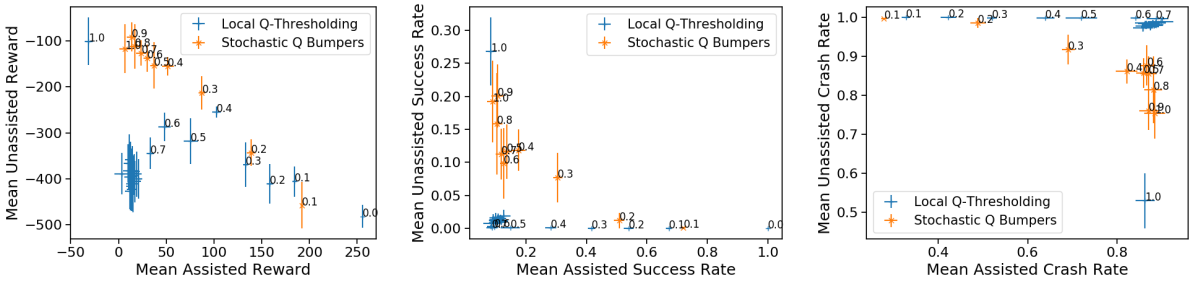
Next, we present results for a more complex task: the Lunar Lander Atari game (Figure 3), which is a simulated flight control task that was tested by Reddy, Dragan, and Levine (2018). The goal of this task is to execute a series of actions (*NOOP*, *fire left engine*, *fire main engine*, *fire right engine*) that results in landing successfully at the goal (which results in $+100$ reward). Crashing or flying out of bounds results in -100, firing the engine results in small negative reward, and moving closer to the goal results in a positive shaping reward. The maximum episode length is 1000 timesteps. We again use the OpenAI Gym domain implementation. To better simulate a single task setting, we modified the Lunar Lander game to use the same initial random seed each episode.

To obtain optimal Q^* values, we trained an agent using Double DQN (van Hasselt, Guez, and Silver 2016) with the same architecture as in (Reddy, Dragan, and Levine 2018) (using a Multi-Layer Perceptron with 2 hidden layers of 64 units each to approximate the Q function). We followed the same hyperparameter settings, including setting $\gamma = 0.99$ for training, which we also use for computing $\hat{G}(s_t)$ in Stochastic Q Bumpers. To give the agent a wider range of experience, we also initialized each episode of training with 50 random actions.

For simulated human learners, we used the same DQN-based learner architecture that was used to obtain optimal Q^* values. Like in Cliff Walking, we decayed the value of ϵ for epsilon-greedy exploration over the first 10% of the episodes from 1 to 0.02, where it remains for the last 90% of episodes (following (Reddy, Dragan, and Levine 2018)). We then ran Stochastic Q Bumpers and Local Q-Thresholding using the same learned Q values, sweeping α parameter settings in $[0, 1]$ in 0.1 increments. For each value of α, we ran 10 trials over 2000 episodes, using different random seeds.[5] Since this task requires high precision to accurately land, we found that $\mathrm{AgentAction}$ with $\mathrm{SECOND} = \mathrm{true}$ does not provide a sufficient level of performance support; we present results with this flag set to false.

Figure 4 shows our results for Lunar Lander. Again, we plot macro averages with 95% confidence intervals based on standard error of the mean. Overall, Stochastic Q Bumpers results are above and to the right of Local Q-Thresholding, indicating improved learning support (given a certain level of performance support) and performance support (given a certain level of learning support). Notably, Stochastic Q Bumpers results in levels of learning support close to the best observed learning (no assistance, Local Q-Thresholding with $\alpha = 1$) for large increases in performance. These large gains in assisted performance with high levels of learning can also be seen in high unassisted success rates (Figure 4b) and low unassisted crash rates (Figure 4c). We were unable to find any settings of α for Local Q-Thresholding (other than completely unassisted $\alpha = 1$) that had learning support values close to these values. In an attempt to locate points with higher learning support close to the unassisted human, we drilled down on 9 additional evenly-spaced values of α between 0.9 and 1 and another 9 between 0.99 and 1. However, these efforts failed to produce any major gains; the resulting values (with low levels of learning support) are shown in the cluster of unlabeled Local Q-Thresholding points.

[5] We selected 2000 episodes as the minimum number of episodes that enabled an unassisted learner to "solve" the task (obtain average reward above 200).



(a) Reward (b) Success Rate (c) Crash Rate

Figure 4: On the Lunar Lander task, Stochastic Q Bumpers significantly outperforms Local Q-Thresholding, both in terms of performance support given a learning objective (fixing a point on the y-axis, Stochastic Q Bumpers is higher on the x-axis), and in terms of learning support given a performance objective (fixing a point on the x-axis, Stochastic Q Bumpers is higher on the y-axis). Measures of learning (unassisted) vs. performance (assisted) are shown in terms of (a) episode rewards, (b) success rate (landing at the goal), and (c) crash rate (crashing the lander). (c) shows inverse learning and performance values. Values of α are annotated for both methods (the dense unlabeled cluster at the bottom left are 18 values of $\alpha \in (0.9, 1.0)$ for Local Q-Thresholding that fail to improve learning).

5 DISCUSSION

Interestingly, only Stochastic Q Bumpers appears capable of increasing performance support without sacrificing large amounts of learning. Across both domains, even small amounts of performance support based on Local Q-Thresholding appear to have very negative effects on learning (as seen by the large gap learning gap between the unassisted agent and Local Q-Thresholding with values of $\alpha < 1$). In the Cliff Walking domain, there is simply no way to set α for Local Q-Thresholding to close this gap and achieve higher levels of learning (we plotted all possible settings). In the Lunar Lander domain, too, extensive drilling down on α values close to 1 also does not result in large learning gains.

As the amount of performance support approaches the maximum level ($\alpha = 0$), both methods show poor learning performance, due to the large amount of misinformation provided to the learner. In the Lunar Lander domain, there is a large phase transition in the amount of support required to successfully land the lander (avoid the -100 penalty) and further to land the lander at the goal (and receive the $+100$ reward). Achieving each of these objectives is difficult and requires careful maneuvering, and failing both of them results in a net loss of 200 reward. Thus, it is not surprising that given the high amount of support required from the agent, little learning is possible in the right half of Figures 4a and 4b (and left half of Figure 4c).

Stochastic Q Bumpers successfully improves performance without sacrificing large amounts of learning in spite of notable challenges we faced in obtaining accurate value estimates for Lunar Lander. Stochastic Q Bumpers in particular relies on accurate Q value estimates in order to compute accurate estimated returns $\hat{G}$ and avoid outcomes that fall below the lower value target determined by v_{min} and α. Unfortunately, we observed that Q values sometimes severely overestimated returns. For example, one timestep before crashing and receiving -100 reward, the Q values are positive. Even with using Double DQN, which is intended to produce less biased value estimates, and trying several other ways of gathering more training samples from the final policy to improve estimates, we were unable to produce more accurate Q values. Stochastic Q Bumpers may benefit further from advances in reinforcement learning that produce more accurate Q values, but it is promising to us that Stochastic Q Bumpers achieves major benefits despite having inaccurate Q values.

In our experiments, we assumed that the assistant (agent) is able to obtain a good policy for the task. In many cases, this will be possible, as agents can learn to perform many tasks, e.g. via simulation or imitation learning even in zero-shot settings (Liu et al. 2018). In other cases, agents may lack important information possessed only by humans, such as the location of a goal (Reddy, Dragan, and Levine 2018). Even in these situations, we are encouraged by the results of Reddy, Dragan, and Levine (2018), which suggest that training Q values with a state space that is augmented with the human action a_h^t can enable the agent learn to decode information from the human's action and take good actions. We are optimistic that these augmented Q values will enable Stochastic Q Bumpers to achieve high levels of performance and learning support even on tasks with goals unknown to the agent.

6 RELATED WORK

In contrast to this work, current approaches to assisting humans seek exclusively to optimize either current performance ("performance support") or future performance ("learning support").

In terms of performance support, prior work has suggested methods that take actions close to a human's, while also ensuring safe, high-reward outcomes (Broad, Murphey, and Argall 2017; Reddy, Dragan, and Levine 2018; Schwarting et al. 2017). Fern et al. (2014) model the problem of selecting optimal assistant actions as a partially observable Markov decision process (POMDP), where the human's goal is unknown. Reddy, Dragan, and Levine (2018) provide a particularly general formulation, which does not make assumptions about knowledge of dynamics or the goal, and which uses model-free methods that do not rely on optimal control. While their method improved task success rates, we demonstrate in this paper that it can be sub-optimal for also supporting human learning. Crucially, this body of work assumes that the human's policy is fixed and therefore does not seek to provide learning support. Other work considers human adaptation to the assistant (Nikolaidis et al. 2017), but not learning of the underlying task (as in this work).

There is a large body of work on AI and machine learning methods that solely focus on helping humans learn, including intelligent tutoring systems and machine teaching methods (Brown and Niekum 2019; Cakmak and Lopes 2012). We are unaware of any prior AI research that considers the objective of learning support in addition to performance support.

7 CONCLUSION AND FUTURE WORK

In this paper, we have argued for the design of assistive agents that provide Learning-Compatible Performance Support (LCPS), and presented the Stochastic Q Bumpers algorithm, which significantly improves performance without sacrificing much learning (and vice versa). We demonstrated the effectiveness of the algorithm over a state-of-the-art shared autonomy algorithm (Reddy, Dragan, and Levine 2018) for a natural class of learners (those that learn from their own actions (Ho, Littman, and Austerweil 2017)), across two domains, including a complex flight control domain. We provided detailed analysis demonstrating limitations of Local Q-Thresholding in the presence of large negative rewards; reductions in regret by Stochastic Q Bumpers in terms of learning and performance of 51% and 43%, respectively, in the Cliff Walking domain; and major benefits to learning and task success rates (Figure 4) in the Lunar Lander domain.

Much work remains to be done in this nascent area. While we conducted our experiments with simulated humans, we are excited to conduct experiments assisting real people next. Our current method is agnostic to the particular human learning model, but we believe that given accurate models of human learning and task deskilling, one can design algorithms to exploit that knowledge for further gains. We also believe that incorporating other teaching strategies (e.g., undoing (Ho, Littman, and Austerweil 2017)) can make further learning improvement possible. Further, we encourage researchers to investigate which of the several possible shared control learning interpretations (including those outlined in (Ho, Littman, and Austerweil 2017)) are most accurate for humans on a variety of tasks. These findings will help to inform the design of future assistants that provide Learning-Compatible Performance Support.

Acknowledgements

We thank the anonymous reviewers for their constructive feedback. This work was supported in part by a Schmidt Foundation gift, a NSF BIGDATA award, and TAL Corporation.

References

Amir, O., E. Kamar, A. Kolobov, and B. J. Grosz (2016). "Interactive Teaching Strategies for Agent Training". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. IJCAI '16.

Broad, A., T. Murphey, and B. Argall (2017). "Learning Models for Shared Control of Human-Machine Systems with Unknown Dynamics". In: *Robotics: Science and Systems Proceedings*.

Brockman, G., V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba (2016). "OpenAI Gym". In: *arXiv:1606.01540 [cs]*.

Brown, D. S. and S. Niekum (2019). "Machine Teaching for Inverse Reinforcement Learning: Algorithms and Applications". In: *AAAI*.

Cakmak, M. and M. Lopes (2012). "Algorithmic and Human Teaching of Sequential Decision Tasks". In: *AAAI*.

Downey, J. E., J. M. Weiss, K. Muelling, A. Venkatraman, J.-S. Valois, M. Hebert, J. A. Bagnell, A. B. Schwartz, and J. L. Collinger (2016). "Blending of Brain-Machine Interface and Vision-Guided Autonomous Robotics Improves Neuroprosthetic Arm Performance During Grasping". In: *Journal of Neuroengineering and Rehabilitation* 13.1, p. 28.

Fern, A., S. Natarajan, K. Judah, and P. Tadepalli (2014). "A Decision-Theoretic Model of Assistance". In: *Journal of Artificial Intelligence Research* 50, pp. 71–104.

Ho, M. K., M. L. Littman, and J. L. Austerweil (2017). "Teaching by Intervention: Working Backwards, Undoing Mistakes, or Correcting Mistakes?" In: *Cog Sci*.

Koedinger, K. R., J. Kim, J. Z. Jia, E. A. McLaughlin, and N. L. Bier (2015). "Learning Is Not a Spectator Sport: Doing Is Better than Watching for Learning from a MOOC". In: *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*.

Liu, Y., A. Gupta, P. Abbeel, and S. Levine (2018). "Imitation from Observation: Learning to Imitate Behaviors from Raw Video via Context Translation". In: *ICRA '18*.

Nikolaidis, S., Y. X. Zhu, D. Hsu, and S. Srinivasa (2017). "Human-Robot Mutual Adaptation in Shared Autonomy". In: *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*.

Reddy, S., A. D. Dragan, and S. Levine (2018). "Shared Autonomy via Deep Reinforcement Learning". In: *Robotics: Science and Systems*. RSS '18.

Schwarting, W., J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus (2017). "Parallel Autonomy in Automated Vehicles: Safe Motion Generation with Minimal Intervention". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*.

Sun, W., A. Venkatraman, G. J. Gordon, B. Boots, and J. A. Bagnell (2017). "Deeply AggreVaTeD: Differentiable Imitation Learning for Sequential Prediction". In: *International Conference on Machine Learning*.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement Learning: An Introduction*. 2nd. The MIT Press.

Torrey, L. and M. Taylor (2013). "Teaching on a Budget: Agents Advising Agents in Reinforcement Learning". In: *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*. AAMAS '13.

van Hasselt, H., A. Guez, and D. Silver (2016). "Deep Reinforcement Learning with Double Q-Learning". In: *AAAI*.