High Confidence Generalization for Reinforcement Learning

James E. Kostas [1] **Yash Chandak** [1] **Scott M. Jordan** [1] **Georgios Theocharous** [2] **Philip S. Thomas** [1]

Abstract

We present several classes of reinforcement learning algorithms that safely generalize to *Markov decision processes* (MDPs) not seen during training. Specifically, we study the setting in which some set of MDPs is accessible for training. For various definitions of safety, our algorithms give probabilistic guarantees that agents can safely generalize to MDPs that are sampled from the same distribution but are not necessarily in the training set. These algorithms are a type of *Seldonian* algorithm (Thomas et al., 2019), which is a class of machine learning algorithms that return models with probabilistic safety guarantees for user-specified definitions of safety.

1. Introduction

In *reinforcement learning* (RL), it is often desirable for trained agents to be robust to changes in their environments and tasks. For example, users of RL algorithms may wish to train an agent using an imperfect simulation and then deploy the agent in the real world. Unfortunately, trained RL agents are often sensitive to changes in their environment: even slight modifications may catastrophically upset an agent's ability to perform a task (Witty et al., 2018). In this work, we present a class of RL algorithms, called *high confidence generalization algorithms* (HCGAs), that provide probabilistic safety guarantees for agents' performances for environments not necessarily seen during training. These guarantees guard against catastrophic outcomes and ensure that agents can successfully generalize to entire distributions of tasks.

This work focuses on the setting where an RL task is represented by some distribution of *Markov decision processes* (MDPs). We assume that the agent is trained on some set of MDPs (that is, a training set) drawn *independently and identically distributed* (i.i.d.) from this distribution. Our algorithms then provide guarantees regarding an agent's performance on MDPs drawn from the distribution, including MDPs not in the training set.

HCGAs first train using a standard RL algorithm and then perform a *safety test* on the resulting policy. The safety test provides guarantees on the trained agent's performance. While some sophistication can be added to the learning process to account for the nature of the task, in their most basic form, the algorithms presented in this work are agnostic to the policy structure, the RL algorithm used for training, and the hyperparameters of the training algorithm. Also, no assumptions are required about the MDPs or the distribution from which they are drawn. These properties make HCGAs versatile and robust; they can be employed in any setting matching the above description, regardless of the training algorithm and policy representation used for the task.

The contributions of this paper are: **1)** a presentation and analysis of HCGAs, and a proof that they provide the probabilistic guarantees that we claim; **2)** a presentation and analysis of a class of HCGAs which provides guarantees regarding the expected performance on the MDP distribution representing the task; **3)** the proposal and analysis of an extension to the class of HCGAs described above; this extension uses control variates designed for the HCGA setting to improve these algorithms without violating the safety guarantees; **4)** a presentation and analysis of classes of HCGAs with risk-sensitive performance guarantees; and **5)** empirical results from two environment distributions that demonstrate that the safety guarantees hold in practice and that the control variate extension may improve results without violating the safety guarantees.

2. Related Work

Safe policy improvement with baseline bootstrapping (SPIBB) (Laroche et al., 2019) is a class of safe RL algorithms with some similarities to HCGAs. Both classes of algorithms generalize with high confidence to some target MDP or MDPs that may be inaccessible for training. However, the problem setting differs in several ways. Unlike HCGAs, SPIBB algorithms do not have direct access to any environment. Instead SPIBB algorithms have a *baseline* policy which they aim to improve and data (state, action, reward, state tuples) gathered from the target MDP using this baseline policy (some SPIBB algorithms need not have

[1]College of Information and Computer Sciences, University of Massachusetts, Amherst, MA, USA [2]Adobe Research. Correspondence to: James E. Kostas <jekostas@umass.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

direct access to the baseline policy (Simão et al., 2020)). These algorithms use this data to return a policy that is probabilistically guaranteed to match or exceed the performance of the baseline policy in the target MDP. Unlike SPIBB algorithms, HCGAs have direct access to a set of MDPs, may not have any data from any target or "test" MDPs, and create a policy from scratch rather than improving upon a baseline.

In RL, *transfer learning* (TL) is the study of how a policy or other knowledge may be transferred between similar but distinct tasks. The HCGA problem setting and approach falls under the broad category of TL. Taylor & Stone (2009) provide a comprehensive survey of TL techniques for RL.

Much of the RL TL literature proposes RL algorithms designed specifically to leverage the TL setting. Konidaris & Barto (2006) investigate how an RL agent may, over a series of similar tasks, learn a reward-shaping function that speeds up the learning of each individual task. Taylor et al. (2007) investigate how a policy may be transferred between two tasks with known intertask mappings between states and actions. Doshi-Velez & Konidaris (2016) and Killian et al. (2017) develop the formulation of *hidden parameter Markov decision processes* (HiP-MDPs). The HiP-MDP framework provides specialized model-based algorithms and is designed for TL between MDPs that are similar but differ slightly in dynamics. These are just a few of the many papers that propose learning algorithms designed to leverage specific properties of the TL setting.

The class of algorithms introduced in this work is *agnostic* to the specific learning algorithm used for initially training the agent: the RL algorithm could be a classical Q-learning algorithm (Watkins, 1989), or it could be a sophisticated algorithm designed to exploit some other aspect of the TL setting (for example, one of the TL algorithms above).

In TL, the *zero-shot* setting requires agents to perform well on unseen tasks without any training time on these tasks. Several papers discussed above fall partly or entirely within this category. Irpan & Song (2019) analyze this RL generalization setting and propose the *principle of unchanged optimality*, which states that "when designing a generalization benchmark, there should exist a [policy] which is optimal for all MDPs." Oh et al. (2017) discuss zero-shot TL for RL and propose a novel approach involving hierarchical skills.

HCGAs are motivated by, and most intuitively applicable to, the zero-shot setting where the principle of unchanged optimality holds, but they can be leveraged in other settings. For example, HCGAs may be used to provide guarantees of safe-but-suboptimal performance for some MDP distribution in which this principle does not hold. The resulting safe-but-suboptimal policy may then be fine-tuned in some specific application environment, as in the meta-learning setting (Finn et al., 2017).

Wang et al. (2019) study generalization in RL in the setting in which the environment transitions can be viewed as deterministic given some random variables that represent the stochasticity. They derive generalization bounds and guarantees for this setting.

Cobbe et al. (2018), Witty et al. (2018), Zhang et al. (2018), and Song et al. (2020) study the phenomena of generalization and overfitting in RL. While our work does not directly study overfitting, our empirical studies make it evident that when our algorithms cannot produce safe solutions, it is primarily because the agent is overfit to the training set; in some sense, the agent "memorizes" the training task(s) in a way that is not generalizable to other similar tasks. Our work provides algorithm designers and end-users with a principled and safe method of ensuring that their RL algorithms and agents do not overfit and fail to generalize in performance-critical applications.

3. Background and Notation

Consider an MDP, $m = (\mathcal{S}, \mathcal{A}, \mathcal{R}, P, R, d_0, \gamma)$. $\mathcal{S}$ is the set of possible states of the MDP, $\mathcal{A}$ is the set of possible actions, and $\mathcal{R}$ is the set of possible rewards. We assume that $\mathcal{S}$ and $\mathcal{A}$ are finite to simplify notation, but the methods in this paper extend to settings where these sets are infinite and uncountable. An *episode* is a sequence of states, actions, and rewards from time $t = 0$ to an indefinite value of t. The random variables S_t, A_t, and R_t are the state, action, and reward at time t. The distribution of the initial state, S_0, is given by $d_0 : \mathcal{S} \to [0, 1]$, and $\gamma \in [0, 1]$ is a parameter called the reward discount factor. A policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ defines the probability of taking each action in each state. Let $\pi(s, a) \coloneqq \Pr(A_t{=}a | S_t{=}s)$. We define a policy to be parameterized by some θ in some feasible set Θ, such that different values of θ result in different policies. $P{:}\mathcal{S}{\times}\mathcal{A}{\times}\mathcal{S}{\to}[0,1]$ is the *transition function*, defined as $P(s, a, s') \coloneqq \Pr(S_{t+1}{=}s' | S_t{=}s, A_t{=}a)$. $R{:}\mathcal{S}{\times}\mathcal{A}{\to}[0,1]$ is the *reward function*, defined as $R(s, a) \coloneqq \mathbf{E}[R_t | S_t{=}s, A_t{=}a]$.

In the typical RL setting, an agent's goal is find a θ that maximizes the *objective function* $J(\theta) \coloneqq \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | \theta\right]$, where conditioning on θ denotes the use of a policy parameterized by θ. In this work, rather than a single MDP, we consider a distribution of MDPs. For all MDPs, we define the objective for MDP m as $J_m(\theta) \coloneqq \mathbf{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | \theta, m\right]$, where "given θ, m" indicates that the environment is MDP m and that the agent is running the policy parameterized by θ. Without loss of generality and to simplify later derivations, we assume that $J_m(\theta) \in [0, 1]$ for all $\theta \in \Theta$.

Below, we typically denote an individual MDP as M_k,

where k is some integer (for example, M_1), a distribution over some set of MDPs as μ, a set of k MDPs as $M_{1:k}$, and the set of all possible sets of MDPs as $\mathcal{M}$. We use $M_1 \sim \mu$ to denote that a single MDP M_1 is sampled from μ, and $M_{1:k} \sim \mu$ to denote that a set of k MDPs $M_{1:k}$ is sampled i.i.d. from μ. When a set of MDPs has a particular name, we denote it as M_{name} below. For example, we denote a training set of MDPs as M_{train}. We refer to the MDPs in such a set as the [name] MDPs (for example, the "training MDPs").

We define the performance (the objective) on a probability distribution of MDPs, μ, as $J_\mu(\theta) \coloneqq \mathbf{E}[J_{M_1}(\theta)|M_1 \sim \mu]$, and the performance given a finite set of MDPs of size k, $M_{1:k}$, as $J_{M_{1:k}}(\theta) \coloneqq \frac{1}{k}\sum_{m \in M_{1:k}} J_m(\theta)$. We define the random variable representing an *episodic return* for an MDP m as $G_m(\theta) \coloneqq \sum_{t=0}^{\infty} \gamma^t R_t$, where the policy used is parameterized by θ.

Let the MDPs $M_1, \ldots, M_k$ denote the individual MDPs in some finite set of MDPs $M_{1:k}$. For some policy parameterized by θ, we define the sample standard deviation of the expected returns, $\hat{\sigma}_J(\theta, M_{1:k})$, to be the sample standard deviation of the set $\{J_{M_1}(\theta), J_{M_2}(\theta), \ldots, J_{M_k}(\theta)\}$.

4. Problem Statement

The primary goal of high confidence generalization algorithms (HCGAs) is the same as in the standard RL setting: to maximize some objective. Specifically, these algorithms maximize $J_\mu(\theta)$, where μ is some arbitrary distribution of MDPs of interest; however, they do so *safely*. That is, they maximize the objective while guaranteeing that some user-defined safety constraint based on the objective or episodic returns holds. Let $f : \Theta \cup \{\text{NSF}\} \to [0, 1]$ be some *safety function* that measures the performance or episodic returns for some solution in $\Theta \cup \{\text{NSF}\}$ (NSF is defined below). All algorithms in this paper guarantee that

$$\Pr\left(f(\text{output}) \geq j\right) \geq 1 - \delta, \tag{1}$$

where $\delta \in (0, 1)$ is a user-specified probability, $j \in [0, 1]$ is a user-defined safety threshold, and where in this one equation only, "output" $\in \Theta \cup \{\text{NSF}\}$ is a random variable that represents the policy parameters output by our algorithm. This output is defined more formally below.

Any algorithm that provides this probabilistic guarantee is a *Seldonian algorithm* as proposed in Thomas et al. (2019). Seldonian algorithms output models (policies in the RL setting) with probabilistic guarantees that the models are safe for a user-defined safety metric, with any desired probability.

This work considers three definitions of the safety function f: one definition based on the expected objective J_μ (Section 6), one risk-sensitive definition based on the "worst-case" MDPs in μ (Section 8.2), and one risk-sensitive definition based on the "worst-case" episodes (Section 8.3). These definitions of f are formally defined in the sections below. For settings where other definitions of safety might be more appropriate, new HCGAs can be created as needed using the approach outlined in this paper.

Consider the case where a user defines an unreasonable value of δ or an unreasonable safety constraint. For example, if the safety constraint requires $J_\mu(\theta) > 0.9$, where θ parameterizes the policy returned by the algorithm, but $J_\mu(\theta') < 0.9$ for all $\theta' \in \Theta$, then this safety constraint cannot be satisfied. For the algorithm to handle such a case safely, we must give it a means to say "I cannot do that." *No Solution Found* (NSF) is this means.

NSF is the output produced by the algorithm when it does not have sufficient confidence that its "best-guess" candidate solution in Θ (defined formally below) will be safe to return; we always define NSF to be safe. Formally, we define $f(\text{NSF}) \coloneqq j$ for all definitions of f. Notice that these algorithms do not give any guarantees concerning the probability of returning a solution that is not NSF: a (useless) algorithm that always returns NSF would technically satisfy the guarantee above. Note that meeting the safety constraint is *not* the only goal of the algorithm; rather, the algorithm's goal is to maximize the expected performance while meeting the safety constraint. Although the naive always-return-NSF algorithm would satisfy the safety constraint, its performance would be poor in terms of the primary objective.

This formulation means that our approach is limited to applications where it is acceptable for the algorithm to not return a solution. The advantage of our approach is that these algorithms will never violate the probabilistic safety guarantees, even if the user-defined values are impossible to satisfy.

In this paper, we study the problem of finding an HCGA, `alg`, that produces a policy that maximizes the objective function, J_μ, while guaranteeing that (1) holds, for various definitions of f. Let $\mathcal{M}$ be the set of possible sets of MDPs that an HCGA could take as input. Formally, we define an HCGA, `alg`, as the function $\texttt{alg} : \mathcal{M} \to \Theta \cup \{\text{NSF}\}$. We can now use this definition to rewrite (1) more formally: $\Pr[f(\texttt{alg}(M_{\text{acc}})) \geq j] \geq 1 - \delta$, where $M_{\text{acc}} \subset \mathcal{M}$ (this set is defined below and is sampled from μ) is the random input to the algorithm.

5. Algorithm Template

The class of algorithms given in this section leverages the Seldonian framework to tackle a difficult and important problem: how to correctly give high-confidence guarantees of generalization. A summary of these algorithms follows. Let M_{acc} be a set of MDPs accessible to the algorithm, sampled i.i.d. from μ. An HCGA partitions M_{acc} into M_{train} and

M_{safety}; M_{train} is used for training, and M_{safety} is used for a safety test. The ratio of the sizes of these two sets can be viewed as a hyperparameter. The algorithm will satisfy the safety guarantees regardless of the setting of this hyperparameter, but might return NSF less frequently for certain values of this parameter. As a simple heuristic, we partition the data into two sets of equal size in all experiments. Additionally, we assume each set consists of at least two MDPs (to satisfy the requirements of all algorithms below). Next, the HCGA uses an RL algorithm and M_{train} to obtain a trained *candidate* policy, θ_c.

Finally, the algorithm performs a *safety test*: it uses M_{safety} to determine whether or not this policy meets some definition of safety for μ. Specifically, the HCGA uses some *high-confidence bounding function* $b : (\Theta \cup \{\text{NSF}\}) \times \mathcal{M} \times (0,1) \rightarrow [0,1]$. For all definitions of f below, we give one or more definitions of b. **Each of these bounding function definitions, combined with the template below, forms a complete algorithm.** The HCGA uses the bounding function to, with the specified confidence, establish a high-confidence lower bound on the value of $f(\theta_c)$. If the candidate policy is safe with the specified confidence, that policy is returned. Otherwise, the algorithm returns NSF. This general form of HCGAs is given in Algorithm 1.

Algorithm 1 HCGA Template

Input : Feasible set Θ, a set of MDPs M_{acc}, user-defined threshold j, probability $1-\delta$, and high-confidence bounding function b.
Output : $\theta \in \Theta \cup \{\text{NSF}\}$

```
Partition M_acc into two data sets, M_train and M_safety;
Compute a θ_c ∈ argmax_{θ∈Θ} J_{M_train}(θ);
if b(θ_c, M_safety, δ) ≥ j then return θ_c;
else return NSF;
```

For all $\theta \in \Theta \cup \{\text{NSF}\}$ and $\delta \in (0,1)$, if Algorithm 1 takes a bounding function b such that $\Pr(b(\theta, M_{\text{safety}}, \delta) \leq f(\theta)) \geq 1-\delta$, then the algorithm will return a safe result with at least probability $1-\delta$. Formally:

Theorem 1. *If* $\Pr(b(\theta, M_{safety}, \delta) \leq f(\theta)) \geq 1-\delta$, *then*

$$\Pr[f(\texttt{alg}(M_{acc})) \geq j] \geq 1-\delta.$$

Proof. See supplementary material Section A. □

Notice that in practice, `alg` can include stochasticity in the optimization process (for example, stochasticity due to the transition function, policy, etc.). One way to capture this stochasticity is to have the algorithm take a random seed as input. For example, in the case where the seed is an integer, `alg` would be the function $\texttt{alg} : \mathcal{M} \times \mathbb{Z} \rightarrow \Theta \cup \{\text{NSF}\}$. For brevity, we make this random seed input implicit.

In Figure 2 in supplementary material Section D, we provide a concise summary of the four HCGAs that we present and study in this paper.

6. Expected Return HCGAs

In this section, we present a class of HCGAs with safety constraints specifying that the expected performance should be above some threshold. Specifically, the algorithms in this class define the safety function f to be J_μ, and therefore give the following probabilistic guarantee: $\Pr\left(J_\mu(\texttt{alg}(M_{\text{acc}})) \geq j\right) \geq 1-\delta$.

Two examples of high-confidence bounding functions for this definition of safety are below, based on Hoeffding's inequality (Hoeffding, 1994) and Student's t-test (Student, 1908), respectively:

$$b(\theta, M_{\text{safety}}, \delta) := J_{M_{\text{safety}}}(\theta) - \sqrt{\ln(1/\delta)/(2|M_{\text{safety}}|)}, \quad (2)$$

$$b(\theta, M_{\text{safety}}, \delta) := J_{M_{\text{safety}}}(\theta) - \frac{\hat{\sigma}_J(\theta, M_{\text{safety}})\texttt{t}_{1-\delta,|M_{\text{safety}}|-1}}{\sqrt{|M_{\text{safety}}|}}, \quad (3)$$

where the sample standard deviation function $\hat{\sigma}_J$ used in (3) is defined in Section 3, and the $\texttt{t}_{*,*}$ used in (3) represents the inverse cumulative distribution function of the Student's t distribution. The t-test bound represented by (3) will often be tighter than that represented by (2), but the t-test bound requires the assumption that the performances of θ_c for the MDPs in M_{safety} are normally distributed. This assumption may not be reasonable, especially for small values of $|M_{\text{safety}}|$. However, by the central limit theorem, it is often a reasonable assumption for large values of $|M_{\text{safety}}|$.

In Algorithm 3 in supplementary material Section L, we give the algorithm represented by the bounding function defined in (2). This serves as an example of how to apply bounding functions to Algorithm 1 to form a complete HCGA. For all other variants, such as that represented by (3), we provide only the bounding functions.

7. Expected Return HCGAs with Control Variates

In this section, we consider a slightly modified problem setting: one in which **1)** each MDP is parameterized by a *known* set of parameters, p_i, in an arbitrary space, $\mathcal{P}$ (for example, the space of possible friction coefficients), and **2)** *parameters* of MDPs can be sampled from the entire distribution of MDPs without needing to (or necessarily having the capability to) construct or run episodes of these MDPs. This setting is of interest when training in simulation using a distribution of MDPs, since the parameters of these MDPs and the MDP distribution, μ, are usually known.

One way to exploit this additional information is to learn a control variate for the candidate policy's expected return given MDP parameters $p_i \in \mathcal{P}$, and to use this control variate to derive unbiased estimates of J_μ that have lower variance than the estimates used in the previous section. These lower variance estimates can then be used in the bounding functions to reduce the probability of returning NSF without compromising the safety guarantees. This method uses a constant which can take any real value; we propose two theoretically grounded methods for choosing optimal values for this constant.

In this section, we write $p_i \in \mathcal{P}$ to denote the parameters of the i^{th} MDP, M_i. Note that $\mathbf{E}[(\text{some expression involving } p_i)|M_i \sim \mu]$ means that p_i are the parameters of MDP M_i.

We introduce a control variate that takes MDP parameters $p_i \in \mathcal{P}$ as input, and estimates the objective value of the corresponding MDP M_i for the policy parameterized by θ. Formally, we write this control variate as the function $\bar{v}_\theta : \mathcal{P} \to [0, 1]$ (recall that we assume the objective and returns are normalized to $[0, 1]$). For example, given some $\theta \in \Theta$ and MDP M_i, $\bar{v}_\theta(p_i)$ estimates $J_{M_i}(\theta)$. The control variate can be an arbitrary function trained with an arbitrary supervised learning algorithm as described below.

Define $Z_i(\theta, c, \bar{v}_\theta, \mu) := J_{M_i}(\theta) - c(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu])$, for some constant $c \in \mathbb{R}$. For brevity, we abbreviate $Z_i(\theta, c, \bar{v}_\theta, \mu)$ as Z_i. Let $M_1, M_2, \ldots, M_k$ be the safety MDPs. While computing the safety test, instead of using $J_{M_1}(\theta), J_{M_2}(\theta), \ldots, J_{M_k}(\theta)$ to estimate the mean, we can instead use the unbiased and potentially lower variance estimates of the mean $Z_1, Z_2, \ldots, Z_k$.

Property 1. *For all $\theta \in \Theta$, for all $c \in \mathbb{R}$, Z_i is an unbiased estimator of $J_\mu(\theta)$.*

The proofs of Properties 1 and 2 and Corollary 1 are given in supplementary material Section B.

Next, we address the question of how to choose a value of c to minimize variance. To estimate an optimal value, we derive an expression for the variance of Z_i and then minimize this expression with respect to c. For the purposes of Property 2 and Corollary 1 below, we assume that the learned control variate is not a constant (that is, it varies with its input, the MDP parameters). This assumption is given more formally in supplementary material Section B.

Property 2.

$$\begin{aligned}
&\operatorname*{argmin}_{c \in \mathbb{R}} \operatorname{Var}(Z_i|M_i \sim \mu) \\
=&\mathbf{E}\Big[(J_{M_i}(\theta) - \mathbf{E}[J_{M_k}(\theta)|M_k \sim \mu]) \\
&\times (\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu])\Big|M_i \sim \mu\Big] \\
&/\ \mathbf{E}\Big[(\bar{v}_\theta(p_{i'}) - \mathbf{E}[\bar{v}_\theta(p_{j'})|M_{j'} \sim \mu])^2\Big|M_{i'} \sim \mu\Big],
\end{aligned}$$

where $\times$ and $/$ denote scalar multiplication and division respectively, split across multiple lines.

An alternative to this method of calculating c follows. Ideally, the control variate $\bar{v}_\theta(p_i)$ will converge to a perfect estimator of $J_{M_i}(\theta)$ given sufficient training data. Consider the value of c for the setting in which the control variate has converged to a perfect estimator: $\bar{v}_\theta(p_i) = J_{M_i}(\theta)$.

Corollary 1. *If $\bar{v}_\theta(p_i) = J_{M_i}(\theta)$, then*

$$\operatorname*{argmin}_{c \in \mathbb{R}} \operatorname{Var}(Z_i|M_i \sim \mu) = 1.$$

Again, the proofs of Properties 1 and 2 and Corollary 1 are given in supplementary material Section B.

Given these properties, it is straightforward to apply control variates to expected value HCGAs such as the Student's t-test HCGA: **1)** After training, before the safety test, evaluate $J_{M_i}(\theta)$ for all $M_i \in M_{\text{train}}$. **2)** Use a supervised learning algorithm of choice to learn a $\bar{v}_\theta(p_i)$, using the $J_{M_i}(\theta)$ values (from the training data). This is a simple regression problem. **3)** Use Property 2 to estimate the optimal c value, using the training MDPs, **or** choose c=1 (we compare and analyze these two methods in supplementary material Section M). **4)** Proceed with the safety test using MDPs $M_1, M_2, \ldots, M_k \in M_{\text{safety}}$ using the set $\{Z_1, Z_2, \ldots, Z_k\}$ instead of the set $\{J_{M_1}(\theta), J_{M_2}(\theta), \ldots, J_{M_k}(\theta)\}$ to calculate the value of the high-confidence bounding function b. These steps are given more formally in Algorithm 2 in supplementary material Section C.

The approach proposed in this section may be particularly advantageous using bounds that vary significantly with the variance of the estimates (for example, Student's t-test). However, even in the case of bounds that do not vary significantly with the variance of the estimates (for example, Hoeffding's), an unbiased but lower variance estimator of the mean can be considered a strict improvement to the HCGAs proposed in the previous section.

8. Risk-Sensitive HCGAs

The above bounds concern the *expected*, or average, return j. In other words, they guarantee that a solution will, with user-specified probability $1 - \delta$, result in an average return greater than or equal to some j. Such solutions could, however, regularly result in MDPs drawn from μ with objective function values less than j or episodes for MDPs drawn from μ with returns less than j. Even a majority of MDPs and/or episodes could result in objective function values and/or returns, respectively, of less than j; as long as the expected return is above j, the criterion above is satisfied.

For this reason, the expected value may not be a suitable measure of safety in some settings. This section proposes

two alternative definitions of safety based on the *conditional value at risk* (CVaR): bounds concerning the *distribution of expected returns* for an MDP drawn from μ, and bounds concerning the *distribution of episodic returns* for an MDP drawn from μ. Supplementary material Section E introduces and defines CVaR for readers unfamiliar with it.

Value at risk (VaR) is another popular risk measure. However, VaR has the disadvantage of being insensitive to rare catastrophic risks (and such risks are one of the primary motivations for definitions of safety not based on the expected value). For this reason, we only introduce CVaR-based HCGAs in this work. However, one could create VaR-based HCGAs for situations where VaR might be a more appropriate measure of safety (see supplementary Section E for a brief discussion and, for readers unfamiliar with it, an introduction to VaR).

8.1. CVaR Bounds

Our algorithms require high-confidence guarantees on the CVaR of a policy's performance or returns, and so we require sample-based bounds on the CVaR of a random variable. In this section, we review such bounds.

We analyze the bounds of Brown (2007) and Thomas & Learned-Miller (2019). The latter bound tended to be tighter in our experiments, so we use it for our results, but the former bound is relatively simple to write and manipulate, and *not* strictly looser, so it may be more desirable in some applications. Therefore, we present the required analyses for both bounds.

Conventions differ as to whether VaR and CVaR are with respect to the lowest or highest possible values of the distribution (Thomas & Learned-Miller, 2019). We use the convention that they are with respect to the lowest possible values, since it better matches the RL setting, where lower values are less desirable. In Section H of the supplementary material, we provide a proof that the "left tail" bound given in Property 3 below is equivalent to the "right tail" bound of Thomas & Learned-Miller (2019). In Sections F and G of the supplementary material, we provide a similar bound and proof for the bounds of Brown (2007).

Let X be a random variable such that $\text{supp}(X) \subseteq [a, b]$. Given a sample of X of size n, let $W_0 := a$, and $W_1, \dots, W_n$ be the order statistics of the sample (that is, the sample sorted into increasing order). Thomas & Learned-Miller (2019) bound CVaR with high confidence:

Property 3. *For all $\delta \in (0, .5]$:*

$$\Pr\left(\text{CVaR}_\alpha(X) \geq W_0 + \frac{1}{\alpha}\sum_{i=1}^{n}(W_{n+1-i} - W_{n-i}) \times \max\left(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha)\right)\right) \geq 1 - \delta,$$

where $\times$ denotes scalar multiplication.

8.2. High Confidence Generalization Across MDPs

Instead of a bound on the value of J_μ, we may instead wish to design an algorithm that, with specified confidence $1 - \delta$, returns a solution with guarantees regarding risk measures for *an MDP drawn from* μ. Such bounds are appropriate when the user desires safety constraints on the expected return for the "worst-case MDPs" in μ. Specifically, for all $\theta \in \Theta \cup \{\text{NSF}\}$, the algorithm in this class uses the following definition of the safety function: $f(\theta) := \text{CVaR}_\alpha(J_{M_1}(\theta)|M_1 \sim \mu)$. Therefore, the probabilistic guarantee is $\Pr(\text{CVaR}_\alpha(J_{M_1}(\texttt{alg}(M_{\text{acc}}))|M_1 \sim \mu) \geq j|M_{\text{acc}} \sim \mu) \geq 1 - \delta$.

Recall that $M_1 \sim \mu$ denotes a single MDP M_1 sampled from μ, and $M_{\text{acc}} \sim \mu$ denotes a set of MDPs M_{acc} sampled i.i.d. from μ. In the inequality above, note that M_1 and M_{acc} are sampled independently of each other.

Given a sample of size n consisting of $J_{M_1}(\theta), \dots, J_{M_n}(\theta)$, where $M_1, \dots, M_n$ are the safety MDPs, let $J_1, \dots, J_n$ be the n order statistics of that sample (that is, the objective values of the n MDPs sorted in increasing order). Define $J_0 := 0$. Applying Property 3, the bounding function is: $b(\texttt{alg}(M_{\text{acc}}), M_{\text{safety}}, \delta) := \frac{1}{\alpha}\sum_{i=1}^{n}(J_{n+1-i} - J_{n-i})\max(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha))$. Recall that, combined with Algorithm 1, this bounding function represents a complete algorithm. We refer to this algorithm as the *CVaR MDP HCGA*.

8.3. High Confidence Generalization for All Episodes

Alternatively, we may instead wish to design an algorithm that, with specified confidence $1 - \delta$, returns a solution with guarantees regarding risk measures for *episodic returns* for episodes drawn from μ. Such bounds are useful when the distribution of *episodic returns* is more relevant to safety than *expected returns*. Specifically, for all $\theta \in \Theta \cup \{\text{NSF}\}$, the algorithm in this class uses the following definition of safety: $f(\theta) := \text{CVaR}_\alpha(G_{M_1}(\theta)|M_1 \sim \mu)$. In other words, users may wish to use this class of algorithms when risk measures on "worst-case episodes" are the best measure of safety. For example, when applying RL to diabetes management (Bastani, 2014), a risk-sensitive measure of episodic returns such as CVaR may be a better safety constraint than the guarantees above: a single bad "episode" could result in

the death of a patient, regardless of the value of the objective function for μ (Section 6) or the objective function for an MDP drawn from μ (Section 8.2).

This is different from the algorithm described in Section 8.2 in that there is no expectation inside the CVaR function: the algorithm considers the returns of individual episodes rather than the expected returns of those episodes (that is, rather than the objective function). The probabilistic guarantee is $\Pr(\text{CVaR}_\alpha(G_{M_1}(\texttt{alg}(M_{\text{acc}}))|M_1 \sim \mu) \geq j|M_{\text{acc}} \sim \mu) \geq 1-\delta$.

Given a sample of size n consisting of $G_{M_1}(\theta)$, $\ldots, G_{M_n}(\theta)$, where $M_1, \ldots, M_n$ are the safety MDPs, let $G_1, \ldots, G_n$ be the n order statistics of that sample. Define $G_0 \coloneqq 0$. The bounding function is $b(\texttt{alg}(M_{\text{acc}}), M_{\text{safety}}, \delta) := \frac{1}{\alpha}\sum_{i=1}^{n}(G_{n+1-i} - G_{n-i}) \max\left(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha)\right)$. In the results, we refer to this algorithm as the *CVaR Episodic HCGA*.

9. Experiments and Results

In this section, we run the four algorithms defined by the bounds above on two sets of MDPs: generalization gridworld and *dynamic arm simulator one* (DAS1) (Blana et al., 2009). In both cases, we define μ to be a uniform distribution over the sets of MDPs. Generalization gridworld is a set of gridworlds in which randomly placed "cliffs" send the agent back to the start state, but a fixed path from the start state to the goal is always clear of cliffs. As a result, while individual MDPs may have many optimal policies, there is only one optimal policy for the entire set.

DAS1 is a detailed and biomechanically plausible human arm simulator with two joints and six muscles. This environment simulates *functional electrical stimulation* (FES) for paralyzed arm muscles; it has been used in biomedical research studying patients suffering from paralysis due to brain or spinal cord injuries (Blana et al., 2009). When patients suffer paralysis due to these types of injuries, the arms and other areas of the body still have the potential to move; the muscles and the local nerves are intact, but the connection to the brain is severed. Advances in the science of FES could someday allow patients paralyzed below the neck to be able to move their arms and other parts of their body again. One challenge of FES is that controllers based on models or trained in simulation often do not perform well when applied to real physiological systems, in part because of the physiological variations between individual subjects (Jagodnik (2014), see Sections 1.4 and 1.5). In fact, individual physiological variations have caused agents trained with DAS1 to not work well on a real-world FES setting with a paralyzed subject (K. M. Jagodnik, personal communication, June 4, 2020). Therefore, the study of the DAS1 domain (and how to ensure that agents trained in it generalize successfully) is an important and impactful application motivating HCGAs. Both environments are described in more detail in supplementary material Section J.

In each experiment, we randomly choose an accessible set of MDPs, M_{acc}, and a *test set*, M_{test}. The latter is a large set of 10,000 MDPs not accessible to the algorithm. We use the test set as the ground truth: it can be used to determine whether an HCGA's returned policy is actually safe, what that policy's true performance is for μ, and what the difference is between its performance for the training set and for μ.

All hyperparameters and experimental details are given in supplementary material Section K.1.

For the plots in this section, we define $J(\text{NSF}) \coloneqq j$ (the same definition as $f(\text{NSF})$). This plotting methodology has the advantage of showing all trials, but has the disadvantage of sometimes causing the HCGA to appear to perform significantly better or worse than the average $J_\mu(\theta_c)$ value, depending on the definition of j and on the frequency with which the algorithm returns NSF. Results excluding trials in which the algorithm returns NSF (and which therefore do not require this definition of $J(\text{NSF})$) are discussed below and are available in supplementary material Section K.3. The HCGAs in this section do not use control variates unless otherwise specified.

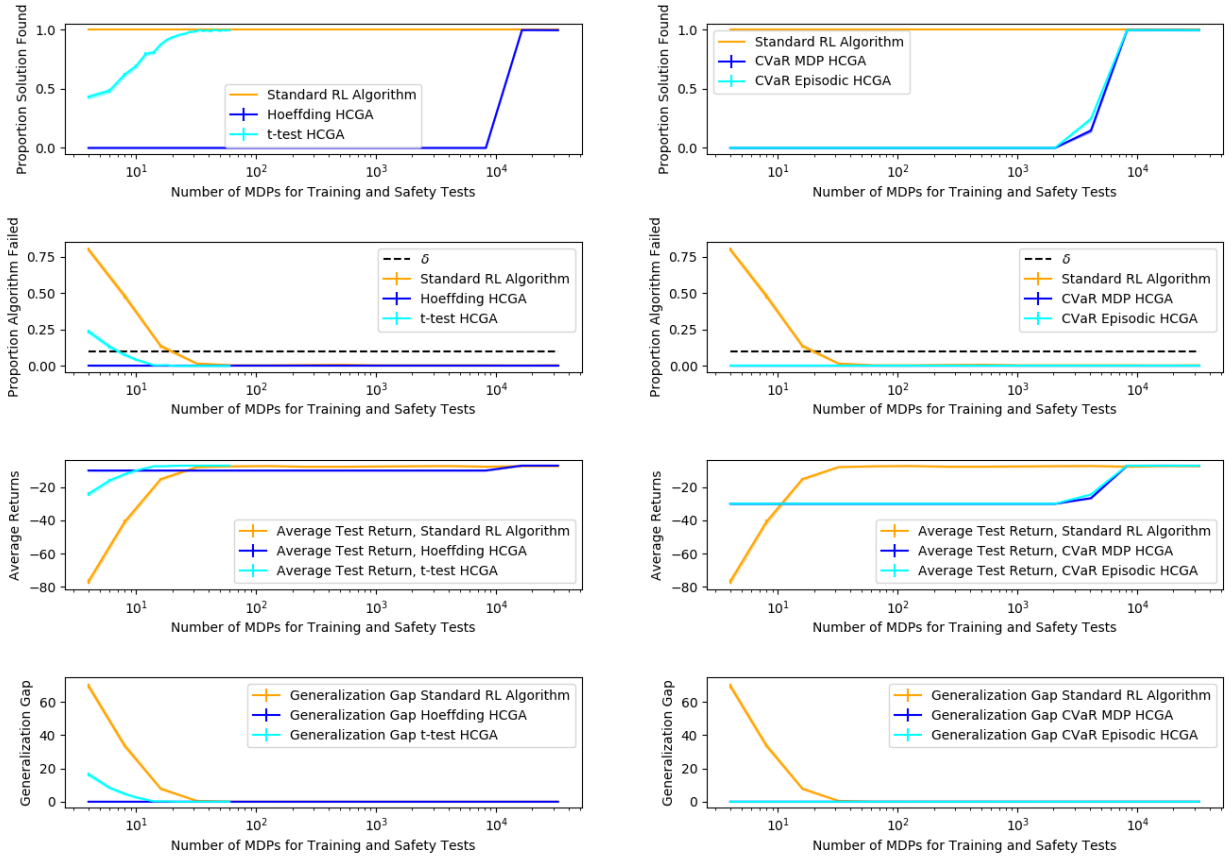
Let the *generalization gap* be defined as the difference between the training and test performances for the algorithm's output $\theta \in \Theta \cup \{\text{NSF}\}$: $J_{M_{\text{train}}}(\theta) - J_{M_{\text{test}}}(\theta)$.

9.1. Generalization Gridworld Results

The results of the generalization gridworld experiments are presented in Figure 1; they demonstrate that the HCGAs' guarantees hold. Notice that the proportion of trials in which the HCGAs failed is below δ in all cases, except in the case of the Student's t-test algorithm for low values of $|M_{\text{acc}}|$. This is expected: at low values of $|M_{\text{acc}}|$, the t-test HCGA's assumption of normality is not reasonable, so the algorithm may return an unsafe solution more than $\delta(100\%)$ of the time. These results demonstrate empirically that the probabilistic guarantees given by HCGAs hold in practice.

The fourth plot in each figure, which plots the generalization gap, makes it evident that HCGAs are preventing overfitting and ensuring generalization: the safety tests detect when a candidate policy is overfit to its training set, and reject that policy as unsafe, resulting in a significantly smaller generalization gap for HCGAs than for standard RL algorithms.

For the Hoeffding and the CVaR HCGAs, notice that the number of MDPs required to reach the best plotted candidate solution (that is, one with a generalization gap near zero) is



(a) Hoeffding and t-test Gridworld Results

(b) CVaR Gridworld Results

Figure 1. Generalization gridworld results. In all plots, the horizontal axis is the number of MDPs accessible for training and safety tests (that is, $|M_{\text{acc}}|$). All error bars represent standard error. In all plots, the phrase "Standard RL Algorithm" represents an algorithm which does not run a safety test, and instead naively maximizes the objective. The first (top) plots show the proportion of trials in which a solution is found (that is, trials in which the algorithm did not return NSF). The second plots in these figures show the proportion of trials in which the algorithm fails; that is, the proportion of trials in which an unsafe solution is returned. In all experiments, $\delta = 0.1$. The third plots in each figure show the average returns. The fourth plots show the generalization gap. These plots were generated using 1000 trials per data point (that is, 1000 trials for each location on the horizontal axis). All other details are discussed in supplementary material Section K.1.

orders of magnitude less than the number of MDPs required to consistently return a solution. This indicates that our simple heuristic of partitioning the data into two sets of equal size is poor in these settings. By allocating fewer MDPs to the training set and more MDPs to the safety set, one could design an HCGA utilizing these bounds that requires significantly fewer MDPs in M_{acc} to return a solution.

9.2. DAS1 Results

DAS1 results using the same layout as Figure 1 are given in Section K.2 of the supplementary material. Individual plots for each of the eight experiments (four HCGAs run for two environments) are given in Section K.3 of the supplementary material (these individual plots use roughly four pages of space, but may be easier to read in the individual format).

The DAS1 results are similar to the gridworld results and demonstrate empirically that 1) the probabilistic guarantees given by HCGAs hold in practice and 2) HCGAs prevent overfitting and ensure generalization.

Notice that in both environments, the two HCGA CVaR algorithms return nearly identical results. Because $J(\theta) = \mathbf{E}[G(\theta)]$, this phenomenon may be common in settings for which variances in episodic returns for each MDP are low, but for which variances in episodic returns across the distribution of MDPs are high. Those conditions will cause the two CVaR definitions of f to take approximately the same value. Future work will study this phenomenon further, but the experiments make it clear that the safety guarantees hold for both CVaR algorithms.

9.3. Control Variate Results

We also study the effect of control variates on expected value HCGAs. Empirical results confirm our theoretical analysis: control variates reduce the variance of the mean estimators without violating the HCGA safety constraints. For more details, see supplementary material Section M.

9.4. Applicability to Computationally Expensive Settings

Because our plots require many trials (100 or 1000 per location on the horizontal axis in our experiments) to reasonably show the "proportion solution found" and "proportion algorithm failed" plots, we chose to perform experiments using environments that are relatively computationally inexpensive. However, when *applying* the HCGA framework to a real-world problem, one must only perform one trial (not hundreds or thousands as in our plots), which makes these algorithms scalable and practicable for computationally expensive applications. Because of our theoretical results, one can confidently apply HCGAs in these settings; the theoretical results hold whether the function approximator is a simple Q-Table, a linear approximator, or the latest and largest deep network architecture. Furthermore, since the computational bottleneck tends to be training (the safety test requires only evaluation of the candidate policies and is thus relatively inexpensive), HCGAs are typically not significantly more computationally expensive than running a standard RL algorithm without the HCGA framework.

10. Conclusion

In this paper, we introduce high confidence generalization algorithms, prove that the probabilistic guarantees given by these algorithms hold, extend one class of these algorithms with control variates, and show empirically that these guarantees hold in practice. Future work will study new types of HCGAs as well as HCGAs in the extrapolation setting, in which M_{acc} is not drawn from the same distribution as M_{test}.

Acknowledgements

Research reported in this paper was sponsored in part by a gift from Adobe, NSF award #2018372, and the DEVCOM Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196 (ARL IoBT CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

References

Bastani, M. Model-free intelligent diabetes management using machine learning. Master's thesis, University of Alberta, 2014.

Blana, D., Kirsch, R. F., and Chadwick, E. K. Combined feedforward and feedback control of a redundant, nonlinear, dynamic musculoskeletal system. *Medical & Biological Engineering & Computing*, 47(5):533–542, 2009.

Brown, D. B. Large deviations bounds for estimating conditional value-at-risk. *Operations Research Letters*, 35(6): 722–730, 2007.

Cobbe, K., Klimov, O., Hesse, C., Kim, T., and Schulman, J. Quantifying generalization in reinforcement learning. *arXiv preprint arXiv:1812.02341*, 2018.

Doshi-Velez, F. and Konidaris, G. Hidden parameter Markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *IJCAI'16: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 1432–1440, 2016.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Hoeffding, W. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pp. 409–426. Springer, 1994.

Irpan, A. and Song, X. The principle of unchanged optimality in reinforcement learning generalization. In *ICML 2019 Workshop on Understanding and Improving Generalization in Deep Learning*, 2019.

Jagodnik, K. M. *Reinforcement learning and feedback control for high-level upper-extremity neuroprostheses*. PhD thesis, Case Western Reserve University, 2014.

Killian, T. W., Daulton, S., Konidaris, G., and Doshi-Velez, F. Robust and efficient transfer learning with hidden parameter Markov decision processes. In *Advances in Neural Information Processing Systems*, pp. 6250–6261, 2017.

Konidaris, G. and Barto, A. Autonomous shaping: Knowledge transfer in reinforcement learning. In *International Conference on Machine Learning*, pp. 489–496, 2006.

Konidaris, G., Osentoski, S., and Thomas, P. Value function approximation in reinforcement learning using the Fourier basis. In *Twenty-fifth AAAI Conference on Artificial Intelligence*, 2011.

Laroche, R., Trichelair, P., and Des Combes, R. T. Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning*, pp. 3652–3661, 2019.

Oh, J., Singh, S., Lee, H., and Kohli, P. Zero-shot task generalization with multi-task deep reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2661–2670, 2017.

Simão, T. D., Laroche, R., and Combes, R. T. d. Safe policy improvement with an estimated baseline policy. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1269–1277, 2020.

Song, X., Jiang, Y., Tu, S., Du, Y., and Neyshabur, B. Observational overfitting in reinforcement learning. In *International Conference on Learning Representations*, 2020.

Student. The probable error of a mean. *Biometrika*, 6(1): 1–25, 1908.

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 2018.

Taylor, M. E. and Stone, P. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(Jul):1633–1685, 2009.

Taylor, M. E., Stone, P., and Liu, Y. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(1):2125–2167, 2007.

Thomas, P. and Learned-Miller, E. Concentration inequalities for conditional value at risk. In *International Conference on Machine Learning*, pp. 6225–6233, 2019.

Thomas, P. S., da Silva, B. C., Barto, A. G., Giguere, S., Brun, Y., and Brunskill, E. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468):999–1004, 2019.

Wang, H., Zheng, S., Xiong, C., and Socher, R. On the generalization gap in reparameterizable reinforcement learning. In *International Conference on Machine Learning*, pp. 6648–6658, 2019.

Watkins, C. *Learning From Delayed Rewards*. PhD thesis, University of Cambridge, England, 1989.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Witty, S., Lee, J. K., Tosch, E., Atrey, A., Littman, M., and Jensen, D. Measuring and characterizing generalization in deep reinforcement learning. *arXiv preprint arXiv:1812.02868*, 2018.

Zhang, A., Ballas, N., and Pineau, J. A dissection of overfitting and generalization in continuous reinforcement learning. *arXiv preprint arXiv:1806.07937*, 2018.

A. Proof of HCGA Guarantees

Theorem 1. *If* $\Pr(b(\theta, M_{safety}, \delta) \leq f(\theta)) \geq 1 - \delta$*, then*

$$\Pr[f(\texttt{alg}(M_{acc})) \geq j] \geq 1 - \delta.$$

Proof. In this proof, we will show that for all θ_c in Θ, $\Pr(f(\texttt{alg}(M_{\text{acc}})) \geq j | \Theta_c = \theta_c) \geq 1 - \delta$, and hence that $\Pr(f(\texttt{alg}(M_{\text{acc}})) \geq j) \geq 1 - \delta$, where $\Theta_c \in \Theta$ is the random variable representing the candidate policy in Algorithm 1. We consider two possible cases: **1)** when $f(\Theta_c) \geq j$ and **2)** when $f(\Theta_c) < j$. In the first case $f(\texttt{alg}(M_{\text{acc}})) \geq j$ always since either $\texttt{alg}(M_{\text{acc}}) = \theta_c$ and by assumption $f(\Theta_c) \geq j$, or $\texttt{alg}(M_{\text{acc}}) = \text{NSF}$ and by definition $f(\text{NSF}) = j$. Hence, $\Pr(f(\texttt{alg}(M_{\text{acc}})) \geq j | \Theta_c = \theta_c) = 1 \geq 1 - \delta$.

Next consider the second case. In this case, we have that for all $\theta_c \in \Theta$ such that $f(\theta_c) < j$:

$$\begin{aligned}
\Pr\left(f(\texttt{alg}(M_{\text{acc}})) \geq j \middle| \Theta_c = \theta_c\right) &\overset{\text{(a)}}{=} \Pr\left(\texttt{alg}(M_{\text{acc}}) = \text{NSF} \middle| \Theta_c = \theta_c\right) \\
&\overset{\text{(b)}}{=} \Pr\left(b(\Theta_c, M_{\text{safety}}, \delta) < j \middle| \Theta_c = \theta_c\right) \\
&\overset{\text{(c)}}{\geq} \Pr\left(b(\Theta_c, M_{\text{safety}}, \delta) \leq f(\theta_c) \middle| \Theta_c = \theta_c\right) \\
&= \Pr\left(b(\theta_c, M_{\text{safety}}, \delta) \leq f(\theta_c) \middle| \Theta_c = \theta_c\right) \\
&\overset{\text{(d)}}{=} \Pr\left(b(\theta_c, M_{\text{safety}}, \delta) \leq f(\theta_c)\right) \\
&\overset{\text{(e)}}{\geq} 1 - \delta,
\end{aligned}$$

where **(a)** follows because when the candidate solution is unsafe (that is, when $f(\Theta_c) < j$), $f(\texttt{alg}(M_{\text{acc}})) \geq j$ if and only if $\texttt{alg}(M_{\text{acc}}) = \text{NSF}$; **(b)** follows from lines 3 and 4 of Algorithm 1, which indicate that $\texttt{alg}(M_{\text{acc}}) = \text{NSF}$ if and only if $b(\theta_c, M_{\text{safety}}, \delta) < j$; **(c)** follows because we are considering the second case, wherein $f(\theta_c) < j$; **(d)** follows because M_{safety} and Θ_c are statistically independent random variables due to Θ_c being computed solely from M_{train}, which is statistically independent of M_{safety}, (that is, for all $M_{1:k} \in \mathcal{M}$, $\Pr(M_{\text{safety}} = M_{1:k} | \Theta_c = \theta_c) = \Pr(M_{\text{safety}} = M_{1:k})$); and **(e)** follows from the assumption in the theorem statement that for all $\theta \in \Theta$, $\Pr(b(\theta, M_{\text{safety}}, \delta) \leq f(\theta)) \geq 1 - \delta$. □

B. Expected Return HCGAs with Control Variates Proofs

For all proofs in this section, recall that $\mathbf{E}[(\text{some expression involving } p_i) | M_i \sim \mu]$ means that p_i are the parameters of MDP M_i (and that therefore p_i itself is random).

Property 1. *For all* $\theta \in \Theta$*, for all* $c \in \mathbb{R}$*,* Z_i *is an unbiased estimator of* $J_\mu(\theta)$*.*

Proof.

$$\begin{aligned}
\mathbf{E}[Z_i | M_i \sim \mu] =& \mathbf{E}\Big[J_{M_i}(\theta) - c\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j) | M_j \sim \mu]\Big) \Big| M_i \sim \mu\Big] \\
=& \mathbf{E}[J_{M_i}(\theta) | M_i \sim \mu] - c(\mathbf{E}[\bar{v}_\theta(p_i) | M_i \sim \mu] - \mathbf{E}[\bar{v}_\theta(p_j) | M_j \sim \mu]) \\
=& \mathbf{E}[J_{M_i}(\theta) | M_i \sim \mu].
\end{aligned}$$

□

Assumption 1 states that the learned control variate varies with its input (that is, that the control variant is not a constant), or, in other words, that the variance is not zero. Formally:

Assumption 1. *For the policy parameterized by* $\theta \in \Theta$*,* $\text{Var}(\bar{v}_\theta(p_i) | M_i \sim \mu) > 0$*.*

Property 2.

$$\operatorname*{argmin}_{c \in \mathbb{R}} \text{Var}(Z_i | M_i \sim \mu) = \frac{\mathbf{E}\Big[\Big(J_{M_i}(\theta) - \mathbf{E}[J_{M_k}(\theta) | M_k \sim \mu]\Big)\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j) | M_j \sim \mu]\Big) \Big| M_i \sim \mu\Big]}{\mathbf{E}\Big[\Big(\bar{v}_\theta(p_{i'}) - \mathbf{E}[\bar{v}_\theta(p_{j'}) | M_{j'} \sim \mu]\Big)^2 \Big| M_{i'} \sim \mu\Big]}.$$

Proof. For brevity, in this proof only, we write $\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu]$ as A, and we write $J_{M_i}(\theta)$ as J. All expectations in this proof are given $M_i \sim \mu$ (written out fully only on the first line), or $M_{i'} \sim \mu$ ($M_{i'}$ instead of M_i to disambiguate in equations where there are more than one of these expectations). For example, $\mathbf{E}[\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu]|M_i \sim \mu]$ is written as $\mathbf{E}[\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu]]$ (or simply as $\mathbf{E}[A]$). Recall from the proof of Property 1 that $\mathbf{E}[A] = 0$, a fact that is exploited in the proof below.

First, we derive an expression for the variance:

$$\begin{aligned}
\mathrm{Var}(Z_i|M_i \sim \mu) =& \mathrm{Var}(J - cA) \\
=& \mathbf{E}[(J - cA)^2] - \mathbf{E}[J - cA]^2 \\
=& \mathbf{E}[J^2] - 2c\mathbf{E}[JA] + c^2\mathbf{E}[A^2] - (\mathbf{E}[J] - c\underbrace{\mathbf{E}[A]}_{=0})^2 \\
=& \mathbf{E}[J^2] - 2c\mathbf{E}[JA] + c^2\mathbf{E}[A^2] - \mathbf{E}[J]^2.
\end{aligned}$$

Minimizing with respect to c by solving for the critical points:

$$\begin{aligned}
0 =& \frac{\partial \mathrm{Var}(Z_i)}{\partial c} \\
=& -2\mathbf{E}[JA] + 2c\mathbf{E}[A^2].
\end{aligned}$$

Next, we verify that this critical point is a minimum. Consider the second derivative, $\frac{\partial^2 \mathrm{Var}(Z_i)}{\partial c^2} = 2\mathbf{E}[A^2]$. $2\mathbf{E}[A^2]$ is positive if $\mathbb{E}[A^2] \neq 0$. $\mathbf{E}[A^2] = \mathbf{E}[(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu])^2] = \mathrm{Var}(\bar{v}_\theta(p_i)|M_i \sim \mu)$. By Assumption 1, $\mathbf{E}[A^2] \neq 0$, so $\mathbf{E}[A^2]$ is positive. Therefore, this critical point is a minimum. Solving for c:

$$\begin{aligned}
c =& \frac{\mathbf{E}[JA]}{\mathbf{E}[A^2]} \\
=& \frac{\mathbf{E}[J(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu])]}{\mathbf{E}[(\bar{v}_\theta(p_{i'}) - \mathbf{E}[\bar{v}_\theta(p_{j'})|M_{j'} \sim \mu])^2]}.
\end{aligned}$$

Consider the numerator of this fraction (for readability, we stop writing all given terms for the remainder of the proof):

$$\begin{aligned}
\mathbf{E}\Big[J\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)]\Big)\Big] =& \mathbf{E}[J\bar{v}_\theta(p_i)] - \mathbf{E}\Big[J\mathbf{E}[\bar{v}_\theta(p_j)]\Big] \\
\overset{\text{(a)}}{=}& \mathbf{E}[J\bar{v}_\theta(p_i)] - \mathbf{E}[J]\mathbf{E}[\bar{v}_\theta(p_j)] \\
=& \mathrm{Cov}(J, v_\theta(p_i)),
\end{aligned}$$

where **(a)** results from the fact that the expectation of J is with respect to M_i, and that M_i and M_j are independent.

The covariance written as $\mathbf{E}[J(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu])]$ is correct but may be numerically unstable, and so it should not be computed in this form. An equivalent and more numerically stable form is:

$$\begin{aligned}
\mathbf{E}[J(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu])] =& \mathrm{Cov}(J, v_\theta(p_i)) \\
=& \mathbf{E}\Big[\Big(J - \mathbf{E}[J]\Big)\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)]\Big)\Big].
\end{aligned}$$

So,

$$c = \frac{\mathbf{E}\Big[\Big(J - \mathbf{E}[J]\Big)\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)]\Big)\Big]}{\mathbf{E}\Big[\Big(\bar{v}_\theta(p_{i'}) - \mathbf{E}[\bar{v}_\theta(p_{j'})|M_{j'} \sim \mu]\Big)^2\Big]}.$$

□

Corollary 1. *If* $\bar{v}_\theta(p_i) = J_{M_i}(\theta)$*, then*

$$\operatorname*{argmin}_{c \in \mathbb{R}} \mathrm{Var}(Z_i|M_i \sim \mu) = 1.$$

Proof. By Property 2,

$$c = \frac{\mathbf{E}\Big[\Big(J_{M_i}(\theta) - \mathbf{E}[J_{M_k}(\theta)|M_k \sim \mu]\Big)\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu]\Big)\Big|M_i \sim \mu\Big]}{\mathbf{E}\Big[\Big(\bar{v}_\theta(p_{i'}) - \mathbf{E}[\bar{v}_\theta(p_{j'})|M_{j'} \sim \mu]\Big)^2\Big|M_{i'} \sim \mu\Big]}.$$

Substituting the control variate for the objective:

$$\begin{aligned} c =& \frac{\mathbf{E}\Big[\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_k)|M_k \sim \mu]\Big)\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu]\Big)\Big|M_i \sim \mu\Big]}{\mathbf{E}\Big[\Big(\bar{v}_\theta(p_{i'}) - \mathbf{E}[\bar{v}_\theta(p_{j'})|M_{j'} \sim \mu]\Big)^2\Big|M_{i'} \sim \mu\Big]} \\ =& \frac{\mathbf{E}\Big[\Big(\bar{v}_\theta(p_i) - \mathbf{E}[\bar{v}_\theta(p_j)|M_j \sim \mu]\Big)^2\Big|M_i \sim \mu\Big]}{\mathbf{E}\Big[\Big(\bar{v}_\theta(p_{i'}) - \mathbf{E}[\bar{v}_\theta(p_{j'})|M_{j'} \sim \mu]\Big)^2\Big|M_{i'} \sim \mu\Big]} \\ =& 1. \end{aligned}$$

□

C. Expected Return HCGAs with Control Variates

Algorithm 2 Expected Return HCGA with Control Variate Template

Input : Feasible set Θ, a set of MDPs M_{acc}, user-defined threshold j, probability $1-\delta$, and high-confidence bounding function b.

Output : $\theta \in \Theta \cup \{\text{NSF}\}$

Partition M_{acc} into two data sets, M_{train} and M_{safety};

Compute a $\theta_c \in \text{argmax}_{\theta\in\Theta} J_{M_{\text{train}}}(\theta)$;

For all $M_i \in M_{\text{train}}$, compute $J_{M_i}(\theta_c)$;

Use the training data collected above (that is, for all $M_i \in M_{\text{train}}$, p_i and $J_{M_i}(\theta_c)$) to compute some $\bar{v}_{\theta_c}$ (this is a regression problem).

Ensure that $\bar{v}_{\theta_c}$ is not a constant (if it is a constant, choose a better function approximator, training or optimization algorithm, and/or control variate hyperparameters; alternatively, use a standard HCGA without a control variate).

Using the whole distribution of MDP parameters from μ, estimate (or calculate exactly if possible) $\mathbf{E}[\bar{v}_{\theta_c}(p_j)|M_j \sim \mu]$. For brevity, define e_v to be the estimate of this expectation: $e_v \coloneqq \mathbf{E}[\bar{v}_{\theta_c}(p_j)|M_j \sim \mu]$;

Estimate an optimal c value: Use the training data to estimate $\mathbf{E}\Big[\Big(J_{M_i}(\theta_c) - \mathbf{E}[J_{M_k}(\theta_c)|M_k \sim \mu]\Big)\Big(\bar{v}_{\theta_c}(p_i) - e_v\Big)\Big|M_i \sim \mu\Big]$ and $\mathbf{E}[(\bar{v}_{\theta_c}(p_{i'}) - e_v)^2|M_{i'} \sim \mu]$, using $J_{\text{train}}(\theta_c)$ to estimate $\mathbf{E}[J_{M_k}(\theta_c)|M_k \sim \mu]$. Use these values (they are the numerator and denominator of the following expression) to estimate

$$c = \frac{\mathbf{E}\big[\big(J_{M_i}(\theta_c) - \mathbf{E}[J_{M_k}(\theta_c)|M_k \sim \mu]\big)\big(\bar{v}_{\theta_c}(p_i) - \mathbf{E}[\bar{v}_{\theta_c}(p_j)|M_j \sim \mu]\big)\big|M_i \sim \mu\big]}{\mathbf{E}\big[\big(\bar{v}_{\theta_c}(p_{i'}) - \mathbf{E}[\bar{v}_{\theta_c}(p_{j'})|M_{j'} \sim \mu]\big)^2\big|M_{i'} \sim \mu\big]}.$$

Alternatively, set $c = 1$;

Define $Z_{i''} \coloneqq J_{M_{i''}}(\theta_c) - c(\bar{v}_{\theta_c}(p_{i''}) - e_v)$. In the bound computation in the next step, for MDPs $M_1, M_2, \ldots, M_k$ in M_{safety}, use $Z_1, Z_2, \ldots, Z_k$ instead of $J_{M_1}(\theta_c), J_{M_2}(\theta_c), \ldots, J_{M_k}(\theta_c)$ to compute $J_{M_{\text{safety}}}(\theta_c)$, $\hat{\sigma}_J(\theta_c, M_{\text{safety}})$, and/or any other relevant statistics;

if $b(\theta_c, M_{\text{safety}}, \delta) \geq j$ then return θ_c;

else return NSF;

Remark: it may be possible to calculate $\mathbf{E}[\bar{v}_{\theta_c}(p_j)|M_j \sim \mu]$ (e_v in the algorithm above) exactly instead of estimating it. For example, if there are finite MDPs in the support of the distribution, and the distribution is uniform over those MDPs, then it

HCGA	Safety Function and Bounding Function	Intuition
Hoeffding	$f(\theta) \coloneqq J_\mu(\theta)$. $b(\theta, M_{\text{safety}}, \delta) \coloneqq J_{M_{\text{safety}}}(\theta) - \sqrt{\ln(1/\delta)/(2\lvert M_{\text{safety}}\rvert)}$.	Safety constraint on the objective.
t-test	$f(\theta) \coloneqq J_\mu(\theta)$. $b(\theta, M_{\text{safety}}, \delta) \coloneqq J_{M_{\text{safety}}}(\theta) - \frac{\hat{\sigma}_J(\theta, M_{\text{safety}}) \mathtt{t}_{1-\delta, \lvert M_{\text{safety}}\rvert - 1}}{\sqrt{\lvert M_{\text{safety}}\rvert}}$.	Safety constraint on the objective.
CVaR MDP	$f(\theta) \coloneqq \text{CVaR}_\alpha(J_{M_1}(\theta) \lvert M_1 \sim \mu)$. $b(\mathtt{alg}(M_{\text{acc}}), M_{\text{safety}}, \delta)$ $\coloneqq \frac{1}{\alpha}\sum_{i=1}^{n}(J_{n+1-i} - J_{n-i}) \max(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha))$.	Safety constraint on the "worst-case MDPs" in μ. This type of HCGA may be useful if a few rare MDPs in $\text{supp}(\mu)$ are suspected to entail catastrophic risks. These HCGAs may also be useful when attempting to transfer a policy to a distribution of MDPs, μ', that is similar to μ (that is, the extrapolation setting). Suppose that, due to the similarity between μ and μ', one can reasonably assume that the performance of the policy for the new setting, $J_{\mu'}(\theta)$, will be no worse than the performance for, e.g., the worst 1% of MDPs sampled from μ. Under this type of assumption, a CVaR MDP HCGA can be straightforwardly applied to inform the user whether the policy is likely to achieve safe performance for μ'.
CVaR Episodic	$f(\theta) \coloneqq \text{CVaR}_\alpha(G_{M_1}(\theta) \lvert M_1 \sim \mu)$. $b(\mathtt{alg}(M_{\text{acc}}), M_{\text{safety}}, \delta)$ $\coloneqq \frac{1}{\alpha}\sum_{i=1}^{n}(G_{n+1-i} - G_{n-i}) \max(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha))$.	Safety constraint on the worst-case episodes. This type of HCGA may be useful if rare episodes may entail catastrophic risk (e.g., the diabetes setting discussed in Section 8.3).

Figure 2. A summary of the four HCGAs that we study in this work.

may be trivial to calculate e_v exactly by computing $\bar{v}_\theta(p_j)$ for every MDP M_j, and taking the mean of the resulting control variate values.

D. HCGA Summary Table

In Figure 2, we provide a summary of the four HCGAs we study in this paper.

E. Background: VaR and CVaR

Value at risk (VaR) is a measure of risk originally developed as a financial metric to quantify how poorly some set of investments might perform, excluding some proportion of worst-case scenarios. Intuitively, for some random variable X and some proportion α, VaR is simply the α-quantile of X. Formally, for some random variable X and some proportion α, we define VaR as:

$$\text{VaR}_\alpha(X) \coloneqq \inf\{x \in \mathbb{R} | \Pr(X \leq x) \geq \alpha\}.$$

Some criticize VaR for being insensitive to catastrophic risks, since it ignores the worst possible outcomes (Brown, 2007).

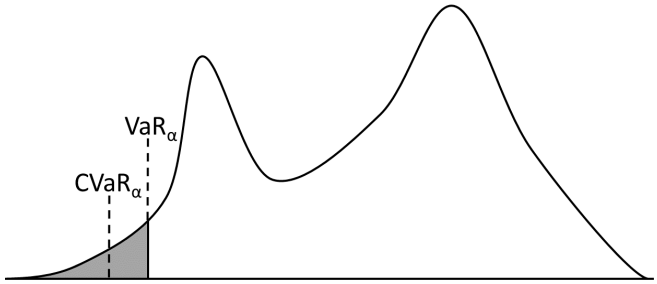


Figure 3. The probability density function of some continuous random variable X. The shaded region has area α. VaR$_\alpha$ is the smallest value such that $\alpha(100\%)$ of samples will be less than it. CVaR$_\alpha$ is the expected value of samples less than or equal to VaR$_\alpha$ (that is, the expected value of samples in the shaded region).

One solution for problems where VaR may not be a suitable measure of risk is *conditional value at risk* (CVaR). Intuitively, given some random variable X and some proportion α, CVaR is the expected value of the lowest α proportion of values of X. In other words, it is the expected value of the "tail" that VaR ignores. Formally, for some continuous random variable X and some proportion α, we define CVaR as:

$$\text{CVaR}_\alpha(X) \coloneqq \mathbf{E}[X|X \leq \text{VaR}_\alpha(X)].$$

For an illustration of VaR and CVaR, see Figure 3.

While this paper restricts itself to CVaR-based HCGAs, one could design VaR-based HCGAs using a high-confidence bound on VaR. Intuitively, VaR-based HCGAs may be appropriate when one wants to ensure that some policy is safe for the majority ($(1-\alpha)100\%$) of MDPs (Section 8.2) or episodes (Section 8.3), but rare, potentially catastrophic risks are either acceptable or nonexistent. CVaR-based HCGAs may be appropriate when one cares less about the overall objective as a measure of safety, but wants to avoid rare catastrophic risks.

F. Brown's CVaR Bound

Let $\hat{C}$ denote the sample-based estimate of CVaR$_\alpha(X)$: $\hat{C} \coloneqq x_{\lceil n\alpha \rceil} - \frac{1}{n\alpha}\sum_{i=1}^{\lceil n\alpha \rceil}(x_{\lceil n\alpha \rceil} - x_i)$, where n is the sample size, and $x_1, ..., x_n$ are the order statistics of the sample (that is, the sample sorted into increasing order). This formulation is equivalent to that of Brown (2007); see supplementary material Section I for the derivation. Brown (2007) bounds CVaR with high confidence:

Property 4. *For all* $\delta \in (0,1)$*, if* $\text{supp}\,(X) \subseteq [a,b]$*:* $\Pr\left(\text{CVaR}_\alpha(X) \geq \hat{C} - (b-a)\sqrt{\frac{5\ln(3/\delta)}{\alpha n}}\right) \geq 1-\delta$*, where* n *is the number of samples of* X *used to calculate* $\hat{C}$*.*

G. Left-Tail Version of Brown's CVaR Bound

Below, we denote the left-tail CVaR that we use as CVaR$_\alpha^L$, and the right-tail CVaR that Brown (2007) used as CVaR$_\alpha^R$. More formally, for a random variable X, we define left and right CVaR respectively, as:

$$\text{CVaR}_\alpha^L(X) \coloneqq \mathbf{E}[X|X \leq \text{VaR}_\alpha^L(X)]$$

and

$$\text{CVaR}_\alpha^R(X) \coloneqq \mathbf{E}[X|X \geq \text{VaR}_\alpha^R(X)],$$

where

$$\text{VaR}_\alpha^L(X) \coloneqq \inf\{x \in \mathbb{R} | \Pr(X \leq x) \geq \alpha\}$$

and

$$\text{VaR}_\alpha^R(X) \coloneqq \sup\{x \in \mathbb{R} | \Pr(X \geq x) \geq \alpha\}.$$

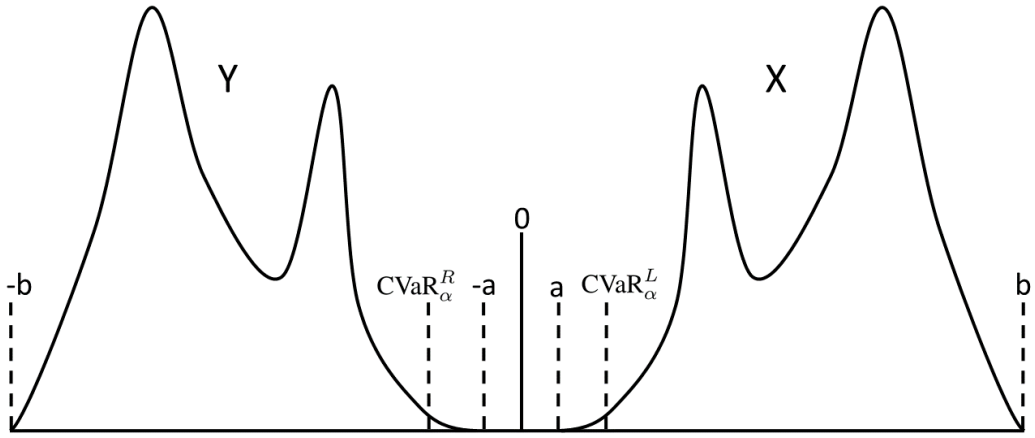


Figure 4. A visualization of the intuition behind the proof of Property 5.

Let $X_1, ...X_n$ be n i.i.d. samples of some continuous random variable X. We denote sample-based estimates of the left and right-tail CVaR values as

$$\hat{C}^L \coloneqq \sup_{x \in \mathbb{R}} \left\{ x - \frac{1}{n\alpha} \sum_{i=1}^{n} \max(0, x - X_i) \right\}$$

and

$$\hat{C}^R \coloneqq \inf_{x \in \mathbb{R}} \left\{ x + \frac{1}{n\alpha} \sum_{i=1}^{n} \max(0, X_i - x) \right\},$$

respectively.

In the proof below, we define another continuous random variable Y, such that $Y \coloneqq -X$. See Figure 4 for an intuitive visualization of this setting. We then these two variables to show that the left- and right-tail bounds are equivalent.

Property 5. *For all* $\delta \in (0, 1)$*, if* $\text{supp}(X) \subseteq [a, b]$*:*

$$\Pr\left(\text{CVaR}_\alpha^L(X) \geq \hat{C}^L - (b-a)\sqrt{\frac{5\ln(3/\delta)}{\alpha n}} \right) \geq 1 - \delta.$$

Proof. Let $Y \coloneqq -X$. Notice that $\text{supp}(Y) \subseteq [-b, -a]$. First, we show that $\text{CVaR}_\alpha^R(Y) = -\text{CVaR}_\alpha^L(X)$:

$$\begin{aligned}
\text{CVaR}_\alpha^R(Y) =& \mathbf{E}[Y | Y \geq \text{VaR}_\alpha^R(Y)] \\
=& \mathbf{E}[Y | Y \geq \sup\{x \in \mathbb{R} | \Pr(Y \geq x) \geq \alpha\}] \\
=& \mathbf{E}[-X | -X \geq \sup\{x \in \mathbb{R} | \Pr(-X \geq x) \geq \alpha\}] \\
=& \mathbf{E}[-X | X \leq -\sup\{x \in \mathbb{R} | \Pr(-X \geq x) \geq \alpha\}] \\
=& -\mathbf{E}[X | X \leq -\sup\{x \in \mathbb{R} | \Pr(-X \geq x) \geq \alpha\}] \\
=& -\mathbf{E}[X | X \leq -\sup\{x \in \mathbb{R} | \Pr(X \leq -x) \geq \alpha\}] \\
=& -\mathbf{E}[X | X \leq \inf\{-x \in \mathbb{R} | \Pr(X \leq -x) \geq \alpha\}] \\
=& -\mathbf{E}[X | X \leq \inf\{x \in \mathbb{R} | \Pr(X \leq x) \geq \alpha\}].
\end{aligned}$$

Applying the left-tail definitions, we get that

$$\begin{aligned}\text{CVaR}_\alpha^R(Y) &= -\,\mathbf{E}[X|X \leq \text{VaR}_\alpha^L(X)] \\ &= -\,\text{CVaR}_\alpha^L(X).\end{aligned}$$

Therefore

$$\text{CVaR}_\alpha^R(Y) = -\text{CVaR}_\alpha^L(X). \tag{4}$$

Next, we show that $\hat{C}_X^L = -\hat{C}_Y^R$. Let $X_1, \ldots, X_n$ be n i.i.d. samples of X, and $Y_1, \ldots, Y_n$ be n i.i.d. samples of Y, such that $Y_1 \coloneqq -X_1, ..., Y_n \coloneqq -X_n$.

$$\begin{aligned}\hat{C}_X^L &\coloneqq \sup_{x\in\mathbb{R}} \left\{ x - \frac{1}{n\alpha}\sum_{i=1}^{n} \max(0, x - X_i) \right\} \\ &= \sup_{x\in\mathbb{R}} \left\{ x - \frac{1}{n\alpha}\sum_{i=1}^{n} \max(0, x + Y_i) \right\} \\ &= -\inf_{x\in\mathbb{R}} \left\{ -x + \frac{1}{n\alpha}\sum_{i=1}^{n} \max(0, x + Y_i) \right\} \\ &= -\inf_{y\in\mathbb{R}} \left\{ y + \frac{1}{n\alpha}\sum_{i=1}^{n} \max(0, Y_i - y) \right\} \\ &= -\,\hat{C}_Y^R.\end{aligned}$$

So

$$\hat{C}_X^L = -\hat{C}_Y^R. \tag{5}$$

Finally, we start with Brown's (2007) right-tail bound for Y:

$$\Pr\left(\text{CVaR}_\alpha^R(Y) \leq \hat{C}_Y^R + ((-a) - (-b))\sqrt{\frac{5\ln(3/\delta)}{\alpha n}} \right) \geq 1 - \delta.$$

Simplifying and applying Equations (4) and (5):

$$\Pr\left(-\,\text{CVaR}_\alpha^L(X) \leq -\hat{C}_X^L + (b - a)\sqrt{\frac{5\ln(3/\delta)}{\alpha n}} \right) \geq 1 - \delta.$$

$$\Pr\left(\text{CVaR}_\alpha^L(X) \geq \hat{C}_X^L - (b - a)\sqrt{\frac{5\ln(3/\delta)}{\alpha n}} \right) \geq 1 - \delta.$$

□

H. Left-Tail Version of Thomas & Learned-Miller's CVaR Bound

In this section, we prove that the left- and right-tail bounds of Thomas & Learned-Miller (2019) are equivalent. Let $X_1, \ldots, X_n$ be n i.i.d. samples of some continuous random variable X, with $\text{supp}(X) \subseteq [a, \infty)$. Let $W_0 \coloneqq a$, and $W_1, \ldots, W_n$ be the order statistics of the sample (that is, $X_1, \ldots, X_n$ sorted in increasing order). We define $\text{CVaR}_\alpha^L(X)$ and $\text{CVaR}_\alpha^R(X)$ as in Section G above. As in the section above, we define another continuous random variable Y, such that $Y \coloneqq -X$.

Property 6. *For all $\delta \in (0, .5]$:*

$$\Pr\left(\text{CVaR}_\alpha^L(X) \geq W_0 + \frac{1}{\alpha}\sum_{i=1}^{n}(W_{n+1-i} - W_{n-i})\max\left(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha)\right)\right) \geq 1 - \delta.$$

Proof. Notice that, since $Y \coloneqq -X$, $\text{supp}(Y) \subseteq (-\infty, -a]$. Let $Y_1, \ldots, Y_n$ be n i.i.d. samples of Y, such that $Y_1 \coloneqq -X_1, \ldots, Y_n \coloneqq -X_n$. Let $Z_1, \ldots, Z_n$ be the order statistics of the sample of Y (that is, $Y_1, \ldots, Y_n$ sorted in increasing order), and let $Z_{n+1} \coloneqq -a$.

Theorem 3 of Thomas & Learned-Miller (2019) states that

$$\Pr\left(\text{CVaR}_\alpha^R(Y) \leq Z_{n+1} - \frac{1}{\alpha}\sum_{i=1}^{n}(Z_{i+1} - Z_i)\max\left(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha)\right)\right) \geq 1 - \delta.$$

In the proof of Property 5 above, we showed that $\text{CVaR}_\alpha^R(Y) = -\text{CVaR}_\alpha^L(X)$. So

$$\Pr\left(-\text{CVaR}_\alpha^L(X) \leq Z_{n+1} - \frac{1}{\alpha}\sum_{i=1}^{n}(Z_{i+1} - Z_i)\max\left(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha)\right)\right) \geq 1 - \delta.$$

Notice that $Z_{n+1} = -a = -W_0$, $Z_n = -W_1, \ldots, Z_2 = -W_{n-1}$, $Z_1 = -W_n$. That is, for $j \in \{1, 2, \ldots, n, n+1\}$, $Z_j = -W_{n+1-j}$.

Applying these equalities:

$$\Pr\left(-\text{CVaR}_\alpha^L(X) \leq -W_0 - \frac{1}{\alpha}\sum_{i=1}^{n}(-W_{n-i} + W_{n+1-i})\max\left(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha)\right)\right) \geq 1 - \delta,$$

so,

$$\Pr\left(\text{CVaR}_\alpha^L(X) \geq W_0 + \frac{1}{\alpha}\sum_{i=1}^{n}(W_{n+1-i} - W_{n-i})\max\left(0, \frac{i}{n} - \sqrt{\frac{\ln(1/\delta)}{2n}} - (1-\alpha)\right)\right) \geq 1 - \delta.$$

□

I. CVaR Estimator Simplification

For some random variable X, given some level $\alpha \in (0, 1)$ and a sample of X of size n, where $x_1, ..., x_n$ are the order statistics of the sample, let $\hat{C}$ denote the sample-based estimate of $\text{CVaR}_\alpha(X)$. We use a definition of $\hat{C}$ that is different from but equivalent to that of Brown (2007); our definition may be more straightforward to implement. We prove that these two definitions are equivalent:

Property 7.

$$\sup_{x \in \mathbb{R}}\left\{x - \frac{1}{n\alpha}\sum_{i=1}^{n}\max(0, x - x_i)\right\} = x_{\lceil n\alpha \rceil} - \frac{1}{n\alpha}\sum_{i=1}^{\lceil n\alpha \rceil}\left(x_{\lceil n\alpha \rceil} - x_i\right).$$

Proof.

$$\begin{aligned}\hat{C} &\coloneqq \sup_{x\in\mathbb{R}}\left\{x - \frac{1}{n\alpha}\sum_{i=1}^{n}\max(0, x - x_i)\right\}\\ &\overset{\text{(a)}}{=} x_{\lceil n\alpha\rceil} - \frac{1}{n\alpha}\sum_{i=1}^{n}\max(0, x_{\lceil n\alpha\rceil} - x_i)\\ &\overset{\text{(b)}}{=} x_{\lceil n\alpha\rceil} - \frac{1}{n\alpha}\sum_{i=1}^{\lceil n\alpha\rceil}\left(x_{\lceil n\alpha\rceil} - x_i\right),\end{aligned}$$

where **a)** follows from the first two steps of the proof of Proposition 4.1 (Brown, 2007) (see below for their reasoning), and **b)** follows from the fact that the order statistics are non-decreasing and $x_{\lceil n\alpha\rceil} - x_i = 0$ for $i = \lceil n\alpha\rceil$. □

A brief elaboration of the reasoning of Brown (2007) for **(a)**: define

$$g(x) \coloneqq x - \frac{1}{n\alpha}\sum_{i=1}^{n}\max(0, x - x_i).$$

Define

$$h(x, i) \coloneqq \begin{cases}0, & \text{if } x_i > x;\\ 1, & \text{if } x_i < x;\\ \text{undefined}, & \text{if } x_i = x.\end{cases}$$

Taking the derivative of g, for all $x \notin \{x_1, \ldots, x_n\}$:

$$\frac{dg(x)}{dx} = 1 - \frac{1}{n\alpha}\sum_{i=1}^{n} h(x, i).$$

Notice that g is continuous and that, except for the n removable discontinuities, $\frac{dg(x)}{dx}$ is monotonically decreasing (including "across" the discontinuities). Therefore, g is concave. Furthermore, notice that for all $x < x_1$, $\frac{dg(x)}{dx} = 1$, and for all $x > x_n$, $\frac{dg(x)}{dx} = 1 - 1/\alpha$, which is negative for $\alpha \in (0, 1)$. More concisely, the derivative switches signs from positive to negative as x increases.

Therefore, $\sup_{x\in\mathbb{R}} g(x)$ will occur either **1)** when $\frac{dg(x)}{dx} = 0$ (or at points at which the left or right derivative is 0, see Figure 5) or **2)** if for all $x \in \mathbb{R}$, $\frac{dg(x)}{dx} \neq 0$, at the removable discontinuity when $\frac{dg(x)}{dx}$ switches from positive to negative (as in Figure 6). By inspection, the point $x = x_{\lceil n\alpha\rceil}$ is the unique $x \in \mathbb{R}$ that satisfies the criteria in both cases.

J. Environment Descriptions

Generalization gridworld is a 5×5 gridworld with deterministic transitions. The reward is -1 at every time step, except for when the agent is in the terminal state, in which case the reward is 0. Each MDP has "cliff" squares which, if entered, send the agent back to the starting position. A single path from the start state to the goal state is clear of cliffs in all MDPs. Specifically, the following sequence of actions is optimal for all MDPs: RIGHT, DOWN, RIGHT, DOWN, RIGHT, DOWN, DOWN, RIGHT. The result is that, while individual MDPs may have many optimal policies, there is only one optimal policy for the entire set of MDPs. The range of possible returns is $[-200, -7]$.

The dynamics and objective of dynamic arm simulator (DAS1) are fully described by Blana et al. (2009). The arm consists of six muscles and two joints. Episodes are of fixed length, and the reward is proportional to the negative square of the distance between the goal and the endpoint of the arm, with a slight penalty proportional to muscle activation. For DAS1, we make the arm and goal initial state in each MDP deterministic and separate possible initial states into 70^4 MDPs (70 possible values of four angles, two of which describe the arm's starting position and two of which describe the goal). We clip the reward at each time step to be in the interval $[-6, 0]$, so that the normalization of the objective function to the range $[0, 1]$ is easier (rewards less than -6 are quite rare, so this does not have much effect).

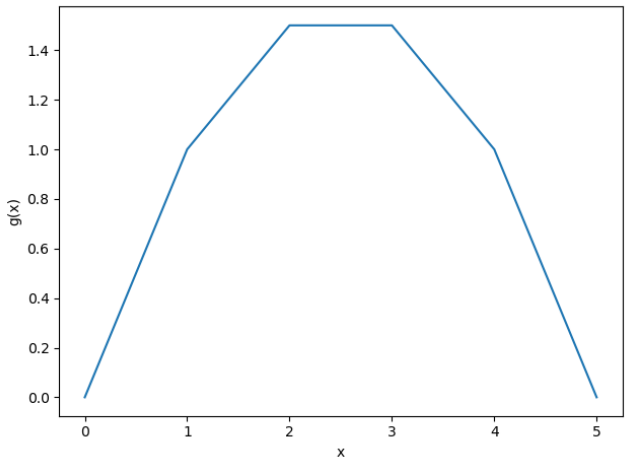


Figure 5. An example of a $g(x)$ for which there exists an x such that $\frac{dg(x)}{dx} = 0$. The supremum is $g(x)$ for all x such that the left and/or right derivatives are 0.

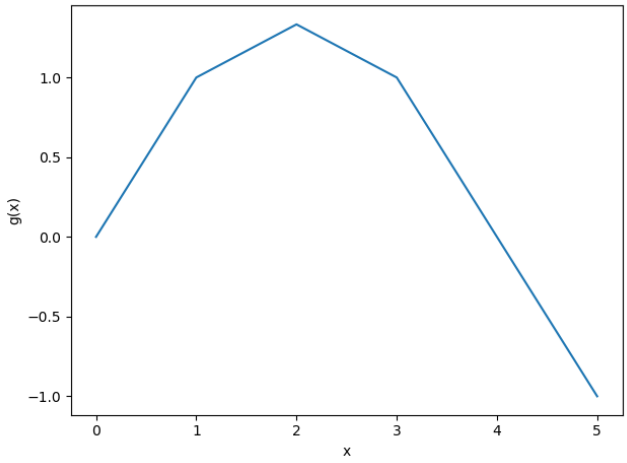


Figure 6. An example of a $g(x)$ for which there does not exist an x such that $\frac{dg(x)}{dx} = 0$. In this case, the supremum lies at the point where $\frac{dg(x)}{dx}$ switches from positive to negative.

K. Full Results and Experimental Details

K.1. Experimental Details

First, we list and discuss the δ, α (for the CVaR quantile, not to be confused with stepsize), and j values used in each experiment. In all experiments, $\delta = 0.1$ and $\alpha = 0.2$.

For the gridworld experiments, $j = -10$ for the Hoeffding HCGA, $j = -8$ for the t-test HCGA (slightly higher than the Hoeffding experiments to highlight the failure behavior with low numbers of MDPs), and $j = -30$ for the CVaR HCGAs. Notice that the CVaR j definitions are significantly lower, since they are for the worst-case tails of the distributions. In Figure 1a above, the standard RL algorithm is plotted using the Hoeffding value, $j = -10$ (the j value affects the plot of the proportion of trials for which the standard RL algorithm failed). A plot of the standard RL algorithm with the t-test value ($j = -8$) is shown in Figure 8b.

For the DAS1 experiments, the Hoeffding and t-test experiments use $j = -25$, and the CVaR experiments use $j = -60$.

In all plots of average returns above, we plot trials which return NSF as j. This choice is because we defined $f(j) \coloneqq \text{NSF}$ and because, intuitively, we have defined NSF to be safe and j is the minimum definition of safe in each experiment. In the plots in Section K.3 below, we provide an alternative interpretation of the same data: excluding NSF trials rather than plotting them as j.

There are four phases to each experiment: **1)** the training phase, in which the candidate policy is trained; **2)** the training evaluation phase, in which the candidate policy's performance is evaluated on the training MDPs; **3)** the safety test phase, in which the policy is run on the safety MDPs, and the safety is test applied; and **4)** the testing phase, in which the candidate policy is run and evaluated on some test set of MDPs. There are always $10{,}000$ test MDPs. For the results to be valid, it is important that sufficient numbers of episodes are run for the training evaluation, safety test, and testing phases.

The number of episodes used for each phase follows. For generalization gridworld, we ran 1024 training episodes per MDP. For DAS1, we ran $10{,}000$ training episodes per MDP.

For the training phase, all MDPs are shuffled into a random order, and each is run once in that order. This process repeats until the maximum number of episodes is run.

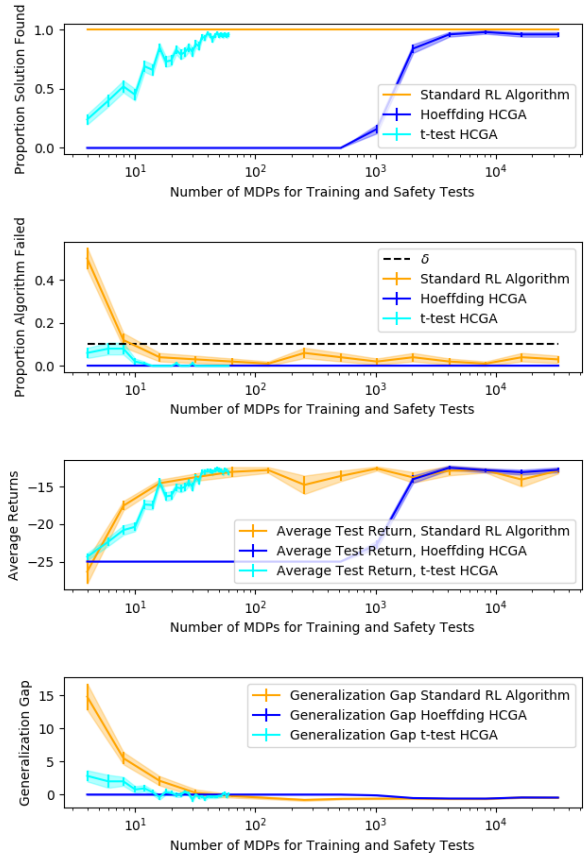
For all experiments, the number of episodes per MDP run in the training evaluation, safety test, and testing phases was $\lceil 10{,}000/n \rceil$, where n is the number of MDPs used in the phase (that is, n is $10{,}000$ for the testing phase, $|M_{\text{train}}|$ for the training evaluation phase, and $|M_{\text{safety}}|$ for the safety testing phase). Notice that, for $n \ll 10{,}000$, this results in approximately $10{,}000$ episodes in the phase. For larger n, this formula also ensures that at least one episode is run for each MDP.

The episodic CVaR HCGA is an exception to the above rule: in the safety test phase, each MDP is only run for one episode, since the safety test samples episodic returns. Sampling a return from each MDP more than once would result in samples not drawn i.i.d. from the distribution of episodic returns (which would invalidate the safety test and the probabilistic safety guarantee).

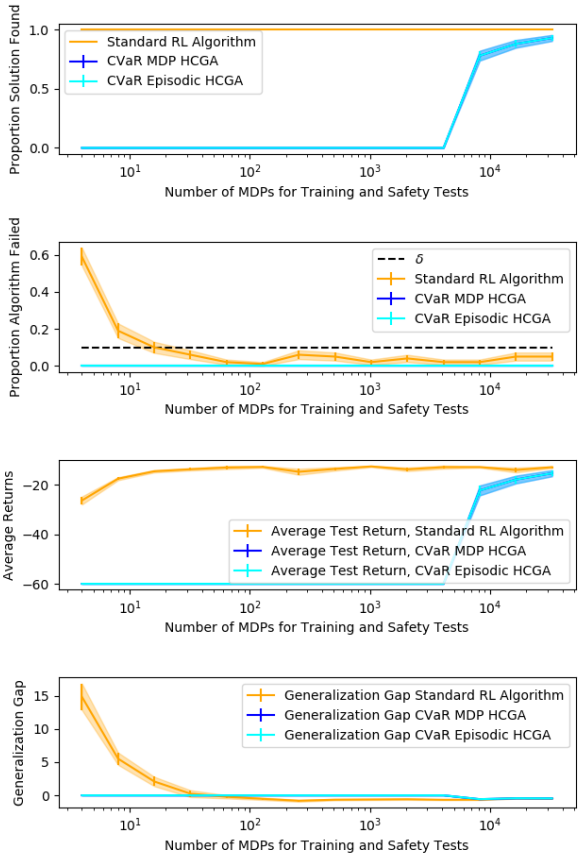
For generalization gridworld, the optimization algorithm used is an actor-critic with eligibility traces (see Sutton & Barto (2018), Section 13.5), a tabular state-action value function, and a softmax policy. The optimization algorithm's hyperparameters were: actor step size = 0.137731127022912, critic step size = 0.31442900745165847, $\gamma = 1.0$, and $\lambda = 0.23372572419318238$. We also experimented with REINFORCE (Williams, 1992), and the outcomes were nearly identical, with all guarantees holding.

For DAS1, the optimization algorithm used was REINFORCE (Williams, 1992), with eligibility traces, a linear function approximator using the Fourier basis (Konidaris et al., 2011), and a softmax policy. The optimization algorithm's hyperparameters were: $\gamma = 1.0$, step size $= 5.736495301650456(10^{-6})$, $\lambda = 0.9082498629094096$, order = 2, and maximum coupled variables = 2.

In practice, one would tune the hyperparameters of the optimization algorithm using the training set (and *not* the safety set). For the purposes of these experiments, we used the entire underlying distribution μ to tune the hyperparameters of the optimization algorithm. This does not break any of our guarantees, since we have access to the entire true underlying distribution. This methodology is also necessary, since, for each trial, the training set is different, and it is not computationally feasible to do a hyperparameter search for each of the hundreds of thousands of trials represented by our eight experiments.



(a) Hoeffding and t-test DAS1 results



(b) CVaR DAS1 Results

Figure 7. See the caption of the Figure 1 for a general description of the plots. These plots were generated using 100 trials per data point. Where the MDP CVaR curves are not visible, they are overlapping with episodic CVaR curves. Notice that the CVaR HCGAs have lower plotted return than the standard RL algorithm. As discussed in supplementary material Section K.1, this is an artifact of plotting the $J(\text{NSF}) = j$. Alternate plots excluding these trials are given in Figure 11 in the supplementary material. These alternative plots show that, excluding NSF trials, the average returns of CVaR HCGAs are higher than those of the standard RL algorithm.

Again, in practice, when one wishes to apply an HCGA and has access to M_{train} and M_{safety}, but not M_{test} or the true distribution, μ, it is important to do hyperparameter tuning *only* on M_{train} and not M_{safety} (otherwise the safety guarantees will be invalid). It is also computationally feasible to do this in practice as this search will only have to be run once (rather than hundreds of thousands of times that would have been required by our experiments).

K.2. DAS1 Results

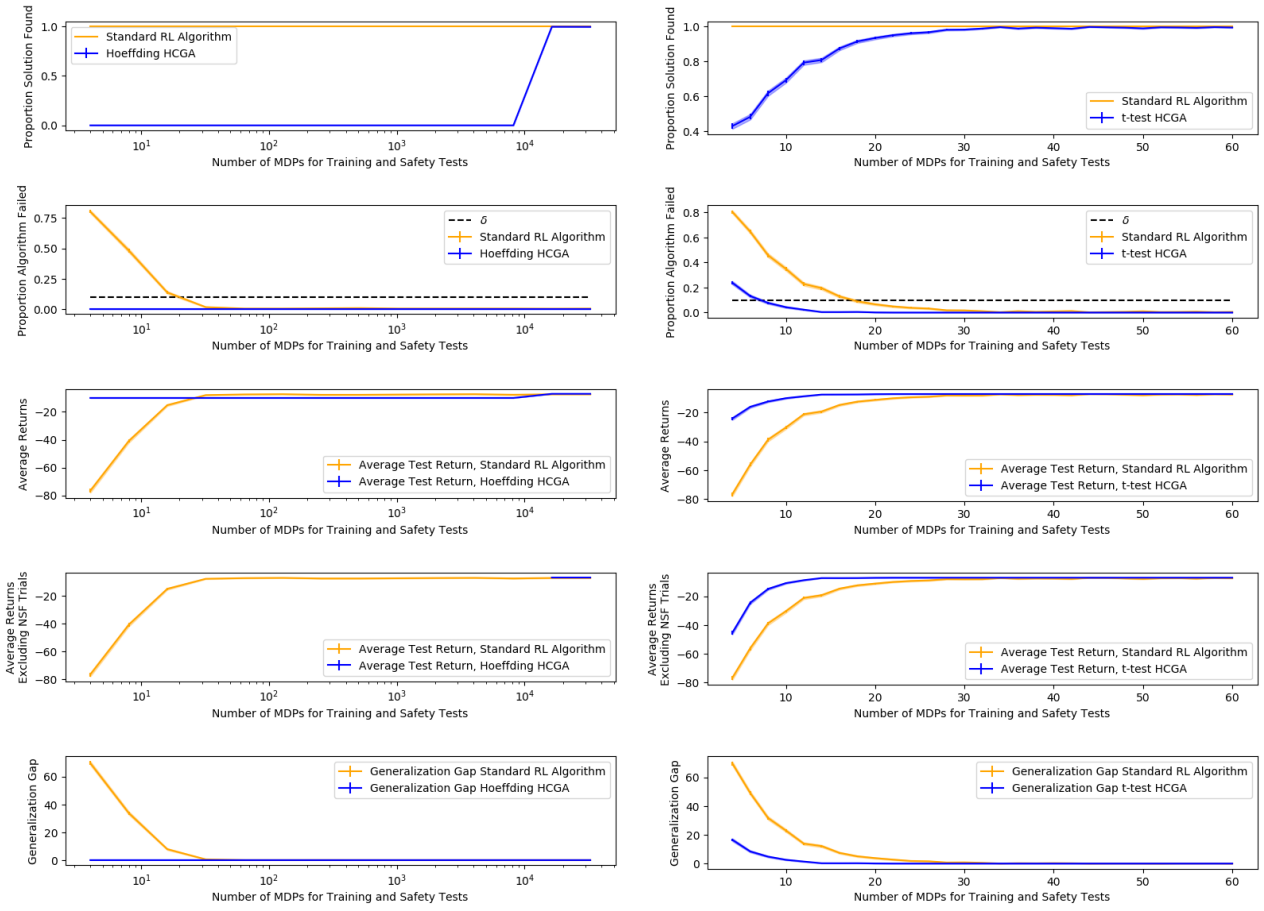
For comparison purposes, the results of the DAS1 experiments in Figure 7 are presented in a layout similar to that of Figure 1 in Section 9. Both environments' results are presented more completely in Section K.3.

K.3. Full Results

In this section, we provide the results for all eight experiments (four HCGAs run for two environments) in eight individual plots. This shows the results of individual experiments more clearly, and allows us to plot the t-test HCGA experiments on a more appropriate linear scale (as opposed to the initial portion of the log scale they are plotted on in Figures 1a and 7a).

We also provide an additional plot for each experiment: the return with NSF trials excluded. That is, instead of plotting the return of NSF trials as j, we exclude those trials from the plot. Notice that, in these alternate plots, some curves do not begin until after $|M_{\text{acc}}|$ is sufficiently large to cause algorithms to return solutions that are not NSF.

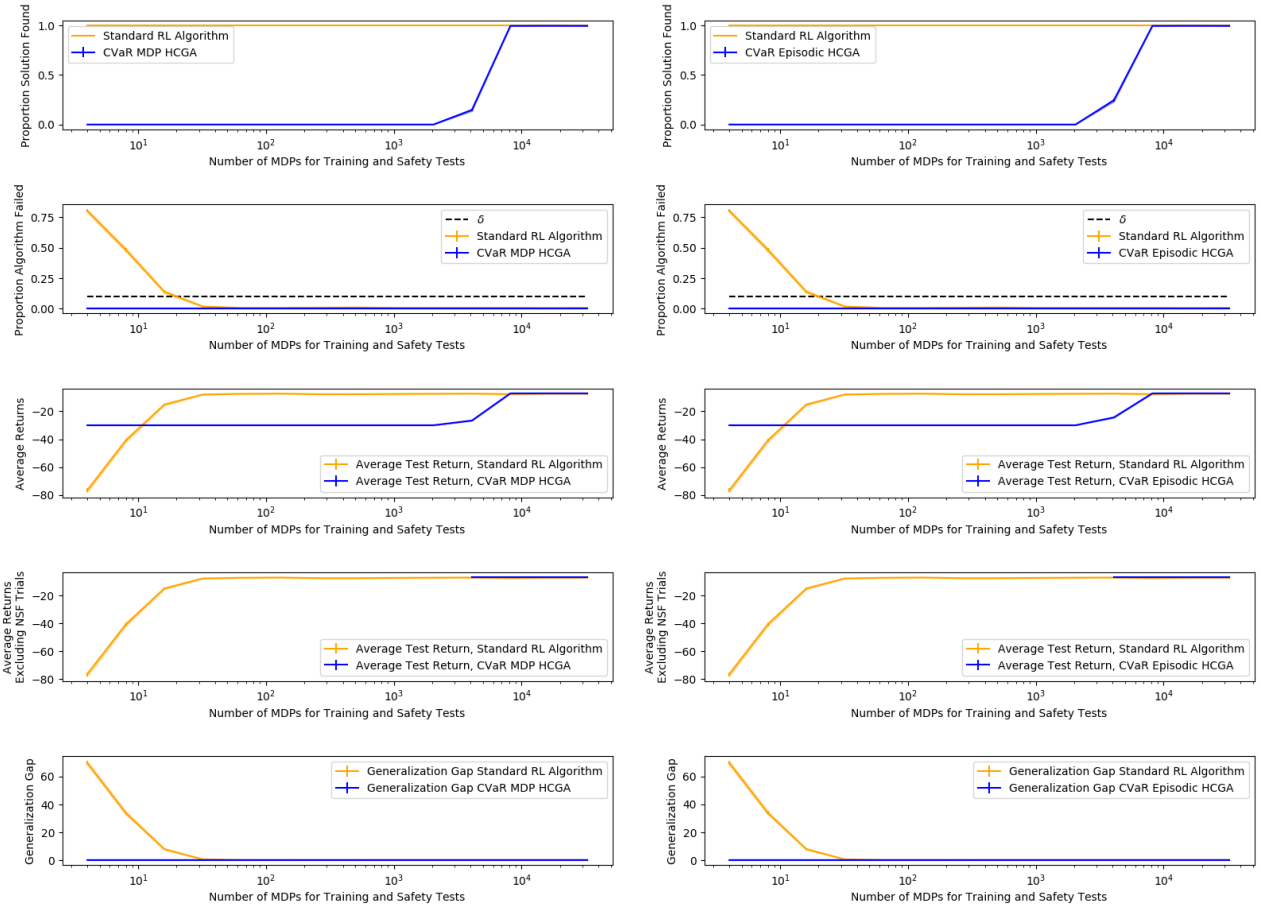
The plots are shown in Figures 8, 9, 10, and 11.



(a) Full Hoeffding Gridworld Results

(b) Full t-test Gridworld Results

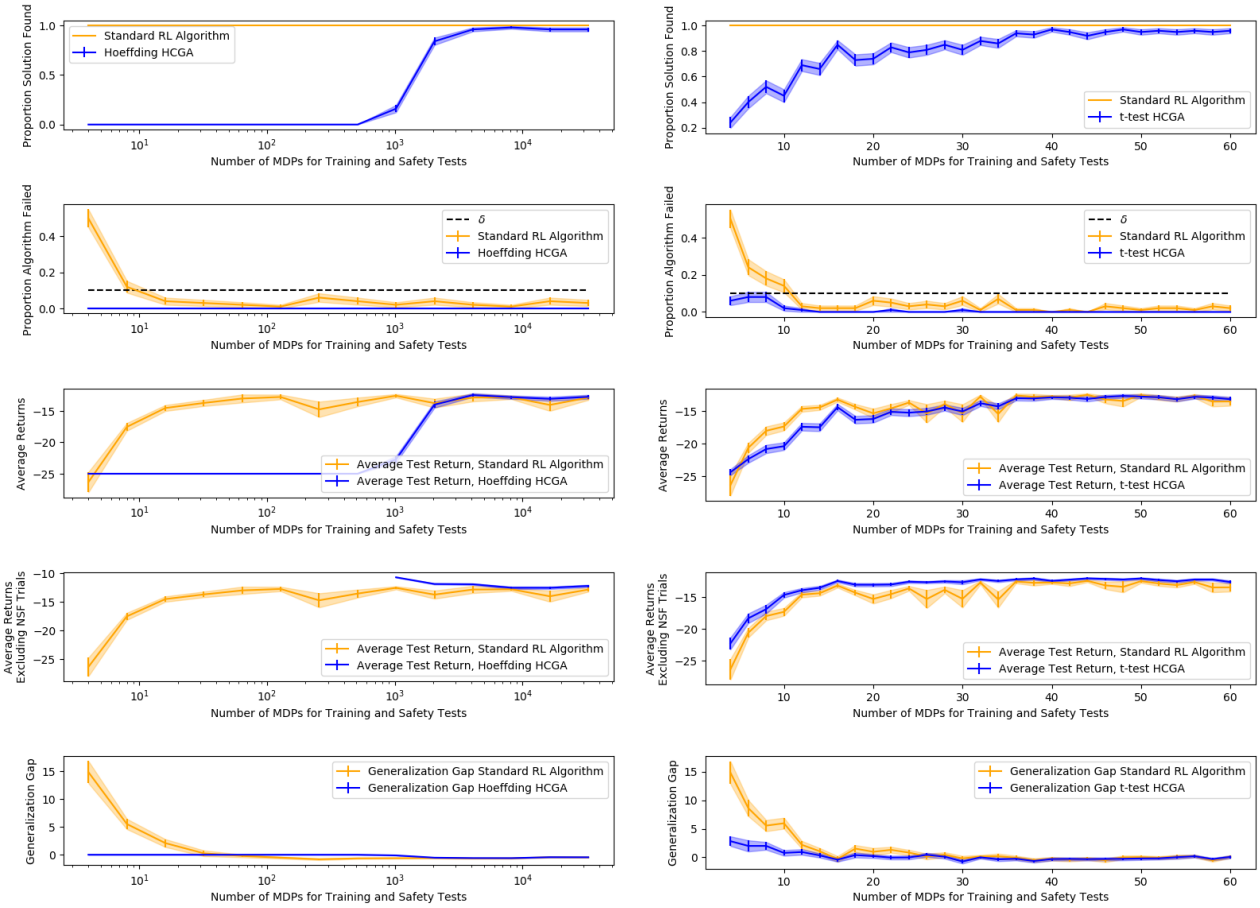
Figure 8. Full Hoeffding and t-test Gridworld Results



(a) Full CVaR MDP HCGA Gridworld Results

(b) Full CVaR Episodic HCGA Gridworld Results

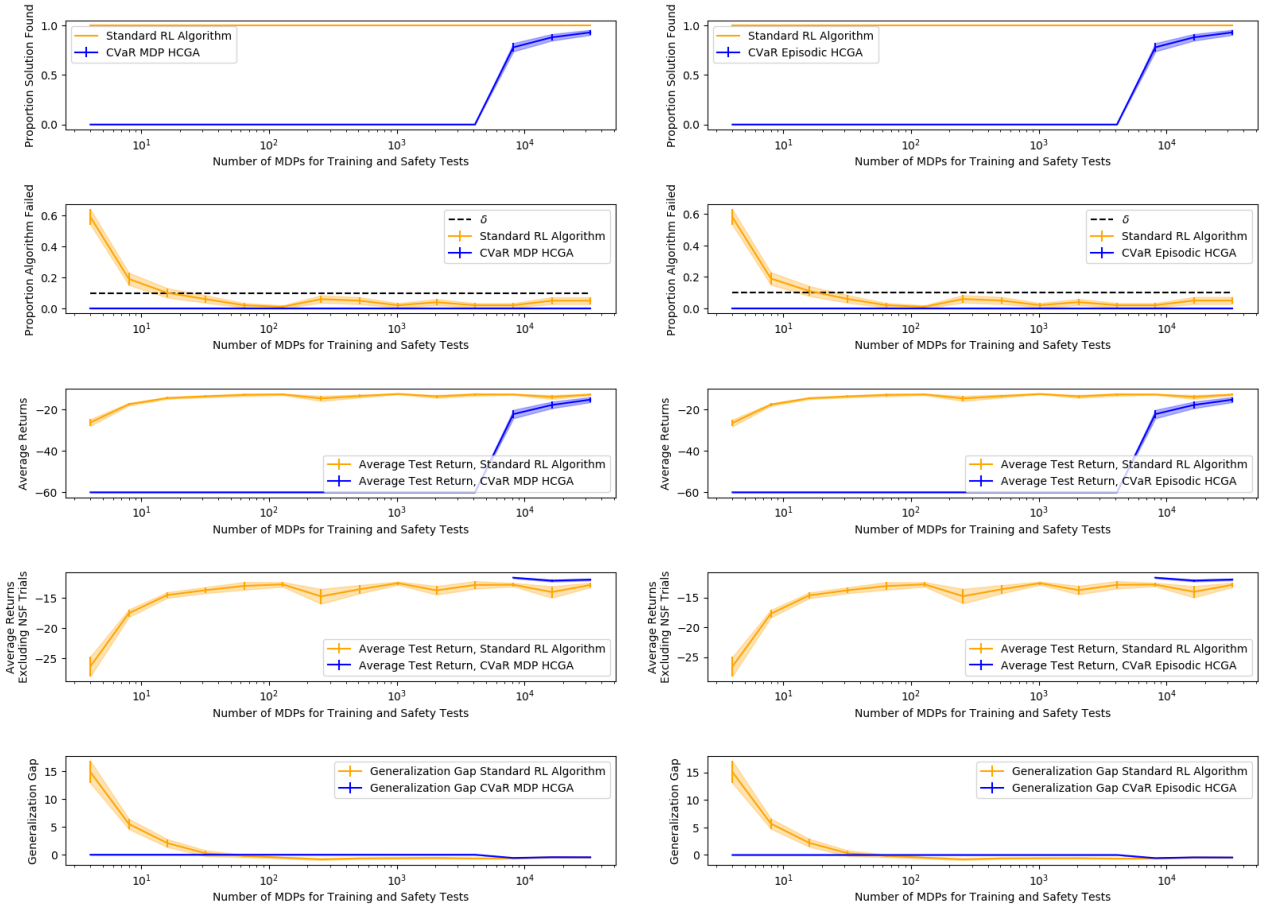
Figure 9. Full CVaR HCGAs Gridworld Results



(a) Full Hoeffding DAS1 Results

(b) Full t-test DAS1 Results

Figure 10. Full Hoeffding and t-test DAS1 Results



(a) Full CVaR MDP HCGA DAS1 Results

(b) Full CVaR Episodic HCGA DAS1 Results

Figure 11. Full CVaR HCGAs DAS1 Results

L. Example HCGA

In this section, we give the algorithm represented by the bounding function defined in (2). This serves as an example of how to apply bounding functions to Algorithm 1 to form a complete HCGA.

Algorithm 3 Expected Return HCGA, Hoeffding Variant

Input : Feasible set Θ, a set of MDPs M_{acc}, user-defined threshold j, and probability $1 - \delta$.
Output : $\theta \in \Theta \cup \{\text{NSF}\}$

Partition M_{acc} into two data sets, M_{train} and M_{safety};
Compute a $\theta_c \in \text{argmax}_{\theta \in \Theta} J_{M_{\text{train}}}(\theta)$;
if $J_{M_{\text{safety}}}(\theta_c) - \sqrt{\frac{\ln(1/\delta)}{2|M_{\text{safety}}|}} \geq j$ then return θ_c;
else return No Solution Found;

M. Expected Return HCGAs with Control Variates: Results and Analysis

In this section, we present and analyze empirical results for the t-test HCGA with control variates. In Sections M.2 and M.3, for procedural gridworld and DAS1 respectively, we demonstrate empirically that the use of control variates with HCGAs reduces the standard deviation of the mean estimates, and that this modification does not violate the HCGAs' safety guarantees. We analyze these results and make a prediction about the kinds of environment distributions for which control

variates will significantly reduce the rate at which HCGAs return NSF. Finally, in Section M.4, we use this prediction to construct and study an MDP distribution. For this MDP distribution, control variates result in a significant decrease in the proportion of trials for which NSF is returned.

M.1. Experiment Details

We only study the t-test HCGA in this section; recall the bounds for the two expected value HCGAs above:

$$\text{Hoeffding: } b(\theta, M_{\text{safety}}, \delta) \coloneqq J_{M_{\text{safety}}}(\theta) - \sqrt{\ln(1/\delta)/(2|M_{\text{safety}}|)}.$$

$$\text{t-test: } b(\theta, M_{\text{safety}}, \delta) \coloneqq J_{M_{\text{safety}}}(\theta) - \frac{\hat{\sigma}_J(\theta, M_{\text{safety}})\text{t}_{1-\delta,|M_{\text{safety}}|-1}}{\sqrt{|M_{\text{safety}}|}}.$$

We study only the t-test HCGA because it has a standard deviation term in the bound that is desirable to minimize, and the Hoeffding HCGA does not have such a term. Ignoring computational cost, using control variates for the Hoeffding HCGA could be considered a strict improvement over not using control variates: control variates will reduce the variance of the mean estimates without compromising the safety guarantees. However, control variates will not usually substantially affect the accuracy of the mean estimate and so cannot be expected to improve the Hoeffding HCGA significantly (unlike the t-test HCGA, which, because of the standard deviation term in the bound, may be substantially improved by control variates).

Recall the two methods for choosing a c value: estimate $c = \frac{\mathbf{E}\left[\left(J_{M_i}(\theta)-\mathbf{E}[J_{M_k}(\theta)|M_k\sim\mu]\right)\left(\bar{v}_\theta(p_i)-\mathbf{E}[\bar{v}_\theta(p_j)|M_j\sim\mu]\right)\middle|M_i\sim\mu\right]}{\mathbf{E}\left[\left(\bar{v}_\theta(p_{i'})-\mathbf{E}[\bar{v}_\theta(p_{j'})|M_{j'}\sim\mu]\right)^2\middle|M_{i'}\sim\mu\right]}$, or choose $c = 1$. Below, we refer to these variants as the *optimal c estimation* variant, and the $c = 1$ variant, respectively. We study both variants below.

In all experiments in this section, $|M_{\text{safety}}| \geq 32$ (so the horizontal axis, $|M_{\text{acc}}|$, begins at $2|M_{\text{safety}}| = |M_{\text{acc}}| = 64$). This is because the t-test bound assumes that the performances of θ_c for the MDPs in M_{safety} are normally distributed. This assumption may not be reasonable, particularly for small values of $|M_{\text{safety}}|$. However, by the central limit theorem, it is often a reasonable assumption for large values of $|M_{\text{safety}}|$.

For simplicity, we use the k-nearest neighbors algorithm for the control variate. We chose $k = 3$ based on intuition, and did not tune or try any other values of this hyperparameter (note that, regardless of the value of this hyperparameter, the safety guarantees will hold and the Z_i values based on the control variate will be unbiased estimators of $J_\mu(\theta)$). As mentioned above, the control variate function approximator, supervised learning algorithm, optimizer, and hyperparameters can be arbitrary. For example, a deep neural network or linear function approximator trained with stochastic gradient descent may also be suitable for many problem settings.

M.2. Control Variates: Generalization Gridworld Results

Consider Figure 12, which shows the generalization gridworld results for HCGAs using control variates. Notice that both variants reduce the standard deviation of the mean estimators compared to the HCGA with no control variate (first plot). However, this fact does not help the HCGAs return more solutions for this environment (third plot), since

$$J_{M_{\text{safety}}}(\theta) \gg \frac{\hat{\sigma}_J(\theta, M_{\text{safety}})\text{t}_{1-\delta,|M_{\text{safety}}|-1}}{\sqrt{|M_{\text{safety}}|}}.$$

In other words, the $\frac{\hat{\sigma}_J(\theta, M_{\text{safety}})\text{t}_{1-\delta,|M_{\text{safety}}|-1}}{\sqrt{|M_{\text{safety}}|}}$ term is already insignificant compared to $J_{M_{\text{safety}}}(\theta)$, so decreasing it more using control variates does not help significantly in practice. Also notice that the optimal c estimates are approximately one (second plot); this fact explains the similar performance of the two control variate variants, and adds empirical support to the theory that $c = 1$ (Corollary 1) is a useful rule. Finally, notice that the control variate HCGAs do not violate the safety guarantees (fourth plot).

As shown in the fourth plot of Figure 12, lowering the standard deviation of the mean estimators for the safety test does not significantly reduce the rate at which the HCGA returns NSF for the generalization gridworld. This is because, for all environments in the distribution, $J(\theta^*)$ is the same value, where θ^* are the parameters of the optimal policy. That is, for all

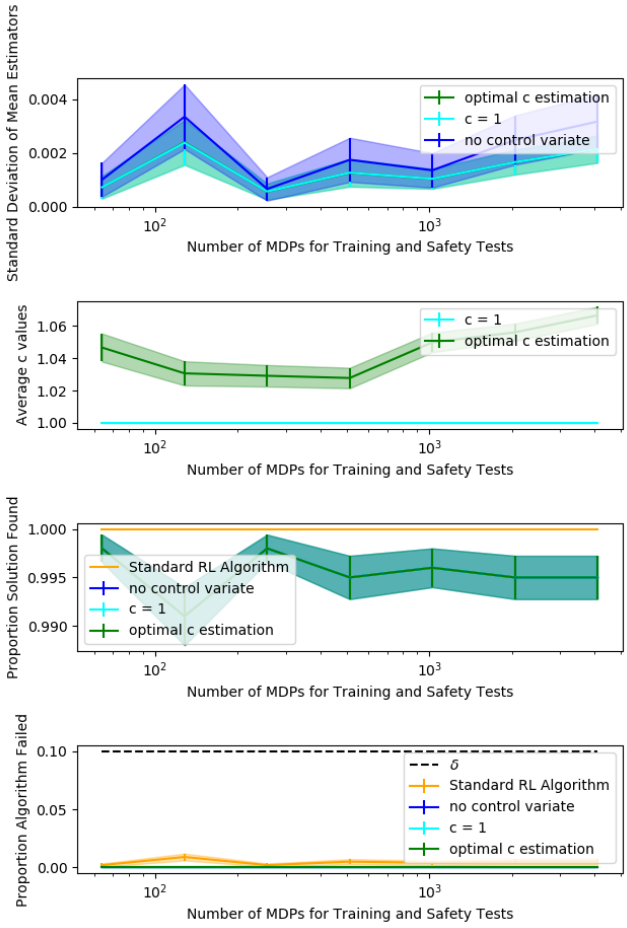


Figure 12. Results for control variates for generalization gridworld. Each location on the horizontal axis corresponds to 1000 trials. Where the $c = 1$ control variate curve is not visible, it is overlapping with the optimal c estimation curve.

MDPs $m \in \text{supp}(\mu)$, $J_m(\theta^*) = -7$ (before return/objective normalization). In this environment, when the HCGA has enough data to pass the safety test, the HCGA tends to learn a policy very close to the optimal policy. Because the standard deviation of the objectives, $\hat{\sigma}_J$, is close to zero, the reduction of the standard deviation of the mean estimates is not helpful in practice for this environment. As we show below, control variates may help more in practice for environment distributions with more variation in the objective functions of their MDPs (given typical candidate policies).

Note, for the generalization gridworld results only: for calculated c values in approximately 0.1% of trials, the denominator of the c calculation is exactly 0, resulting in an undefined value of c. In these rare cases, we set $c = 1$ based on Corollary 1. Notice that, for these rare trials, this method disregards the step in Algorithm 2 that says "ensure that $\bar{v}_{\theta_c}$ is not a constant"; the control variate is a constant in these cases, which results in the calculated c value being undefined. Because our goal is to show properties of these algorithms across many trials (not to apply them to a real-world problem in practice), we use this method for this subsection only. Since this denominator value is never zero in the environments below, this method is only necessary for this generalization gridworld subsection, not for the subsections below. (This is because, given typical candidate policies, the standard deviation between the objectives of different MDPs tends to be at least an order of magnitude lower for generalization gridworld than for the other two environments below. This low standard deviation can sometimes result in a constant control variate, since the function the control variate is supposed to approximate is a constant or nearly a constant.)

M.3. Control Variates: DAS1 Results

Figure 13 shows the DAS1 results for HCGAs using control variates; the results are similar to the generalization gridworld results. Both variants reduce the variance of the mean estimators compared to the HCGA with no control variate. Also notice that, once again, the optimal c estimates are approximately one (second plot), which adds further empirical support to the theory that $c = 1$ (Corollary 1) is a useful rule. Additionally, notice that the control variate HCGAs do not violate the safety guarantees (fourth plot).

Like the results for the generalization gridworld, these results do not show that the control variates result in a significant increase in the proportion of trials in which a solution is found.

Consideration of these results naturally leads to the observation that control variates might be more useful in practice for an environment distribution with much higher standard deviation between the objectives of different MDPs (for optimal or near optimal policies). We explore this observation in the next subsection.

M.4. Control Variates: Stochastic Generalization Gridworld Results

We modified the generalization gridworld MDP distribution so that optimal or near optimal policies would result in higher standard deviation between the objectives of different MDPs: Half of MDPs in the modified distribution are exactly like the MDPs in the unmodified distribution. The other half of MDPs have a stochasticity in their transition functions of 0.5. This means that with probability 0.5, the agent would move as in the normal generalization gridworld transition function. With probability 0.5, the environment will ignore the agent's action, and force it to move in a random direction (or attempt to move in that direction, since the boundaries of the environments or the cliffs may interfere). There are four directions, so the result is that the agent has a 0.625 probability of moving in its "intended" direction, and a 0.125 probability of moving in one of the other three directions. We call this the *stochastic generalization gridworld*.

To account for the changed dynamics, we changed the definition of safety for this environment by decreasing the value of a safe j and increasing the probability with which a safe solution must be returned: $1 - \delta := 0.99$ and $j := -23$ (before return/objective normalization).

As shown in Figure 14, the results match our hypothesis; control variates are more useful in practice for an MDP distribution like the stochastic generalization gridworld. The "proportion solution found" plot shows that, for $|M_{\text{acc}}| = 64$ and $|M_{\text{acc}}| = 128$, the control variates substantially reduce the probability that NSF is returned. As in the experiments above, the safety guarantees were not violated, and the optimal c estimation value was approximately 1.

These results empirically confirm our theory that control variates can reduce the value of the standard deviation of the mean estimates for expected return HCGAs, and that this modification does not violate the HCGA safety guarantees. Furthermore, these results show that the control variate extension may be particularly useful for environments which have high variance in objectives between MDPs (for a typical candidate policy).

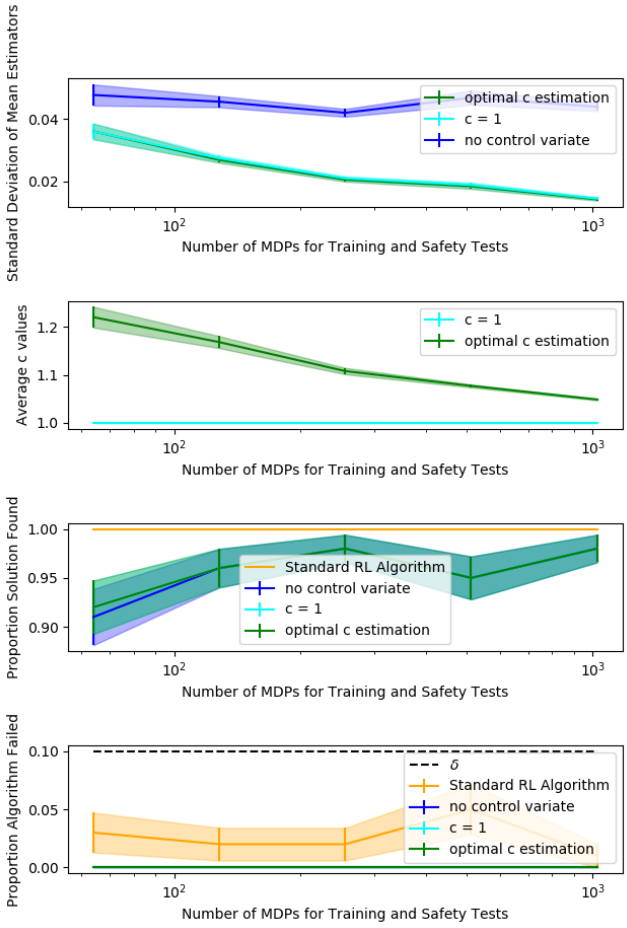


Figure 13. Results for control variates for DAS1. Each location on the horizontal axis corresponds to 100 trials. Where the $c = 1$ control variate curve is not visible, it is overlapping with the optimal c estimation curve.

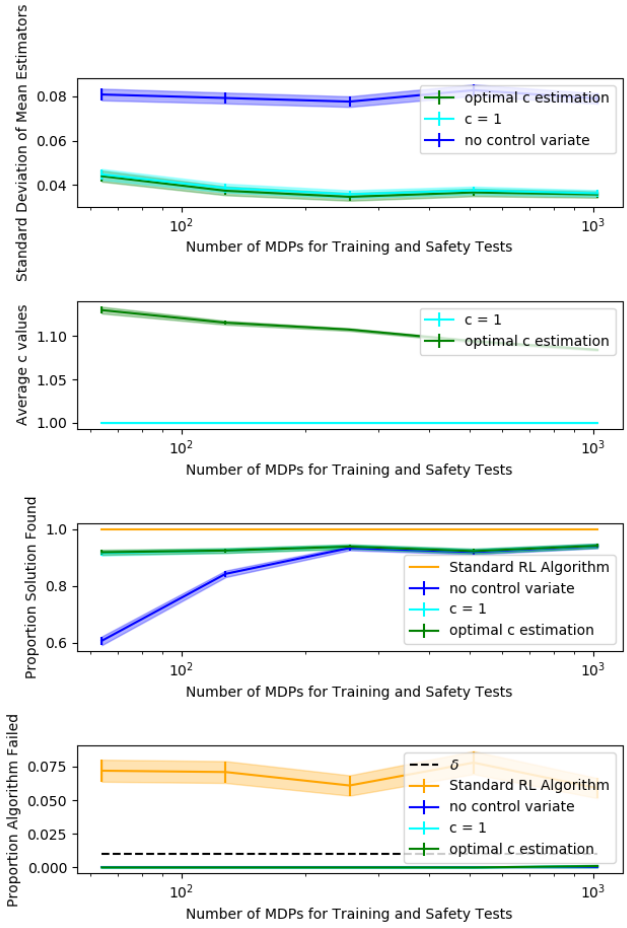


Figure 14. Results for control variates for stochastic generalization gridworld. Each location on the horizontal axis corresponds to 1000 trials. Where the $c = 1$ control variate curve is not visible, it is overlapping with the optimal c estimation curve.