A RISK-SENSITIVE POLICY GRADIENT METHOD

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard deep reinforcement learning (DRL) agents aim to maximize expected reward, considering collected experiences equally in formulating a policy. This differs from human decision-making, where gains and losses are valued differently and outlying outcomes are given increased consideration. It also wastes an opportunity for the agent to modulate behavior based on distributional context. Several approaches to distributional DRL have been investigated, with one popular strategy being to evaluate the projected distribution of returns for possible actions. We propose a more direct approach, whereby the distribution of full-episode outcomes is optimized to maximize a chosen function of its cumulative distribution function (CDF). This technique allows for outcomes to be weighed based on relative quality, does not require modification of the reward function to modulate agent behavior, and may be used for both continuous and discrete action spaces. We show how to achieve an unbiased estimate of the policy gradient for a broad class of CDF-based objectives via sampling, subsequently incorporating variance reduction measures to facilitate effective on-policy learning. We use the resulting approach to train agents with different "risk profiles" in penalty-based formulations of six OpenAI Safety Gym environments, finding that moderate emphasis on improvement in training scenarios where the agent performs poorly generally improves agent behavior. We interpret and explore this observation, which leads to improved performance over the widely-used Proximal Policy Optimization algorithm in all environments tested.

1 INTRODUCTION

While deep reinforcement learning (DRL) has been used to master an impressive array of simulated tasks in controlled settings, it has not yet been widely adopted for high-stakes, real-world applications. One reason for this gap is a lack of distributional perspective in standard artificial agents. Endowing agents with such perspective could make their decision-making more robust, potentially leading to increased safety, increased trust from humans, and more widespread real-world adoption.

In reinforcement learning (RL), *risk* arises due to uncertainty around the possible outcomes of an agent's future actions. It is a result of randomness in the operating environment, mismatch between training and test conditions, and the inherent randomness of a stochastic policy. *Risk-sensitive* policies, or those that consider more than the mean over the distribution of possible outcomes, offer the potential for added robustness under uncertain and dynamic conditions. There is an evolving landscape of algorithmic paradigms for handling risk in RL, from constraint-based approaches originating from optimal control (Chow et al., 2015; Achiam et al., 2017; Zhong et al., 2020) to adversarial approaches emerging from AI Safety (García & Fernández, 2015; Amodei et al., 2016). Within this landscape, learning approaches that optimize distributional measures offer the ability to express design preferences over the full distribution of potential outcomes, through the specification of a risk-sensitivity criterion.

Distributional RL has been studied for value-based methods, with a popular strategy being to estimate the distribution of Q-values for each of member of a discrete set of potential actions through use of the distributional Bellman equation (Bellemare et al., 2017; Dabney et al., 2018a;b). However distributional RL has not been widely explored for policy gradient methods, which could permit direct optimization of risk-sensitive measures and naturally accommodate both discrete and continuous action spaces.

In the following, we introduce a novel framework for risk-sensitive learning using policy gradients. Our approach allows agents to be trained with different risk profiles through the design-time specification of both utility and weight functions, with the latter being defined over the estimated distribution

of full-episode rewards. This framework enables agent-based learning that captures aspects of human decision-making, such as overemphasis of rare occurrences and diminishing marginal-utility relative to a target outcome (Kahneman & Tversky, 1979). It can also be used to implement another key strategy of human learning: emphasizing one's deficiencies. This tendency has been leveraged to expedite off-policy learning (Schaul et al., 2016), through prioritization of memories in a replay buffer. We demonstrate an ability to improve performance in six OpenAI Safety Gym environments (Ray et al., 2019) over the widely-used Proximal Policy Optimization algorithm (Schulman et al., 2017b) through the use of an objective that emphasizes poor outcomes.

2 Related Work

Distributional approaches to risk-sensitive RL have primarily been explored in the value-based setting. Therein, the value distribution has been explicitly modeled through categorical techniques (Bellemare et al., 2017) or quantile regression (Dabney et al., 2018a) and used to improve both value predictions and overall performance. Recent works utilize distributional modeling in the actor-critic setting to enable application in continuous action spaces, again demonstrating improved performance over baseline approaches (Ma et al., 2020; Zhang et al., 2021; Duan et al., 2021). In value-based approaches, risk-sensitivity criteria are applied at run time as a nonlinear warping of the estimated value distribution, in order to increase robustness and improve alignment with human decision-making.

Policy gradient approaches offer similar promise for risk-sensitive RL, but require direct optimization of a parameterized policy with respect to a distributional objective. Some existing methods are limited to a specific class of learning objective, such as the set of concave risk measures that allow a globally-optimal solution (Tamar et al., 2015; Zhong et al., 2020). Others allow a broader class of measures but are more restrictive in the class of policies that can be represented (Prashanth et al., 2016). We aim for a risk-sensitive policy gradient approach that both offers significant flexibility in the choice of learning objective and can learn policies parameterized by a deep neural network.

Various measures have been considered in the context of risk-sensitive RL, including exponential utility (Pratt, 1964), percentile performance criteria (Wu & Lin, 1999), value-at-risk (Leavens, 1945), conditional value-at-risk (Rockafellar & Uryasev, 2000), and prospect theory (Kahneman & Tversky, 1979). In this work, we consider a class of risk-sensitivity measures motivated by Cumulative Prospect Theory (CPT) (Tversky & Kahneman, 1992). CPT uniquely models two key aspects of human decision-making: (1) a utility function u computed relative to a reference point that induces more risk-averse behavior in the presence of gains and (2) a weight function w that prioritizes outlying events. Specific forms of u and w are given in Tversky & Kahneman (1992), but the general form of CPT admits a wide variety of risk-sensitive objectives.

Here we show how to train agents to optimize this class of objectives through sampling-based estimation of their policy gradients and requisite variance reduction. The final algorithm resembles well-known on-policy approaches such as Proximal Policy Optimization (Schulman et al., 2017b) and is similarly widely applicable. Although we do not explore it here, the incorporation of an appropriate risk-sensitivity criterion could additionally enable risk-aware exploration and adversarial training for increased robustness (Pinto et al., 2017; Parisi et al., 2019; Zhang et al., 2020).

3 CDF-Based Policy Optimization

In this section we formalize the class of distributional objectives to be considered, derive a sampling-based approximation of their policy gradients, enact variance reduction on this estimate, and use the result to produce a practical learning algorithm.

3.1 Preliminaries: Problem and Notation

Standard deep reinforcement learning seeks to maximize the expected reward of an agent over encountered trajectories; that is, it maximizes the objective

$$J(\theta) = E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]. \tag{1}$$

Here $p_\theta(\tau)$ is the distribution over trajectories $\tau = \mathbf{s}_1, \mathbf{a}_1, \ldots, \mathbf{s}_T, \mathbf{a}_T$ induced by the policy parameterized by θ, and $\mathbf{s}_t$, $\mathbf{a}_t$, and $r(\mathbf{s}_t, \mathbf{a}_t)$ denote the state, action, and reward at time t, respectively. To

enable the incorporation of distributional context, we instead consider the objective

$$J(\theta) = \int_{-\infty}^{+\infty} u(r(\tau)) \frac{d}{dr(\tau)} \Big(w(P_\theta(r(\tau))) \Big) dr(\tau), \tag{2}$$

where $u(r(\tau))$ is the utility associated with full-trajectory reward $r(\tau) \equiv \sum_t r(\mathbf{s}_t, \mathbf{a}_t)$ and w is a differentiable[1] weighting function of the CDF $P_\theta(r(\tau)) = \int_{-\infty}^{r(\tau)} p_\theta(r') dr'$.

The form in Equation 2 applies to a broad class of distributional objectives. It encompasses the standard expectation value, where the utility function is the identity mapping and the weight function w is linear. It is inspired by CPT (Tversky & Kahneman, 1992), where two integrals of the form (2) are used to model (1) the tendency of humans to be more risk-averse in the presence of gains than losses and (2) our inclination to emphasize the best and worst possible outcomes in our decision-making.

3.2 CDF-Based Policy Gradient

To optimize the objective (2), we first evaluate its gradient with respect to the policy parameters θ. Working toward a representation that can be sampled, we assert the independence of the reward on θ and use the chain rule to write

$$\nabla_\theta J(\theta) = \int_{-\infty}^{\infty} u(r(\tau)) \frac{d}{dr(\tau)} \Big(w'(P_\theta(r(\tau))) \nabla_\theta P_\theta(r(\tau)) \Big) dr(\tau), \tag{3}$$

where w' is the derivative of w with respect to $P_\theta(r(\tau))$. Writing $P_\theta(r(\tau))$ as an integral, using the structure of the underlying MDP, and applying the "log-derivative trick" together yield the following representation of the gradient of the CDF:

$$\begin{aligned} \nabla_\theta P_\theta(r(\tau)) &= \nabla_\theta \int_{-\infty}^{r(\tau)} p_\theta(r') dr' = \nabla_\theta \int_{\tau'} H(r(\tau) - r(\tau')) p_\theta(\tau') d\tau' \\ &= \int_{\tau'} H(r(\tau) - r(\tau')) \nabla_\theta p_\theta(\tau') d\tau' = \int_{\tau'} H(r(\tau) - r(\tau')) p_\theta(\tau') \nabla_\theta \log p_\theta(\tau') d\tau'. \end{aligned} \tag{4}$$

Here H is the Heaviside step function. Alternatively, we could write

$$\nabla_\theta P_\theta(r(\tau)) = \nabla_\theta \Big(1 - \int_{r(\tau)}^{\infty} p_\theta(r') dr' \Big) = - \int_{\tau'} H(r(\tau') - r(\tau)) p_\theta(\tau') \nabla_\theta \log p_\theta(\tau') d\tau'. \tag{5}$$

Either expression, or a combination of the two, may be substituted into (3) and the result sampled over N trajectories by following a scheme similar to that in Prashanth et al. (2016). First, trajectories $i = 1 \dots N$ are sorted in order of increasing reward $r(\tau)$. Then

$$\nabla_\theta J(\theta) \approx \sum_{i=1}^{N} u(r(\tau_i)) \Big(w'\Big(\frac{i}{N}\Big) \nabla_\theta P_\theta(r(\tau_i)) - w'\Big(\frac{i-1}{N}\Big) \nabla_\theta P_\theta(r(\tau_{i-1})) \Big), \tag{6}$$

where the term $w'(0) \nabla_\theta P_\theta(r(\tau_0)) \equiv 0$. $\nabla_\theta P_\theta(r(\tau_i))$ may be sampled in one of two ways, based on either (4) or (5):

$$\nabla_\theta P_\theta(r(\tau_i)) \approx \frac{1}{N} \sum_{j=1}^{i} \sum_{t=1}^{T_j} \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t} | \mathbf{s}_{j,t}) \approx -\frac{1}{N} \sum_{j=i+1}^{N} \sum_{t=1}^{T_j} \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t} | \mathbf{s}_{j,t}). \tag{7}$$

The expression (6) represents an unbiased estimate of the gradient of the distributional objective (2), and may be used to optimize a policy in a manner similar to REINFORCE (Williams, 1992). The policy gradient for expected reward represents a special case, wherein $w' \equiv 1$ and $u(r) = r$.

3.3 Variance Reduction

Unfortunately, as is the case with the policy gradient for expected reward, the expression (6) suffers from high variance. This prevents sample-efficient learning and results in poor convergence. Fortunately, several steps may be taken to reduce the variance of the gradient estimate.

[1] w could also be piecewise-differentiable, if required.

First, note that terms of the form $f(\tau_i, \mathbf{a}_{j,t}, \mathbf{s}_{j,t}) = u(r(\tau_i))\nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}, \mathbf{s}_{j,t})$, while nonzero, do not contribute to the gradient estimate in expectation when $i \neq j$. These terms result from the expression of the gradient for the CDF, but we do not expect to include terms where the utility of one trajectory scales the log-likelihood of another. To evaluate their expectation value, note that

$$\begin{aligned} E_{\tau_i \sim p_\theta(\tau), \tau_j \sim p_\theta(\tau)} f(\tau_i, \mathbf{a}_{j,t}, \mathbf{s}_{j,t}) &= E_{\tau_i \sim p_\theta(\tau), \tau_j \sim p_\theta(\tau)} u(r(\tau_i)) \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \\ &= E_{\tau_i \sim p_\theta(\tau)} \left[u(r(\tau_i)) E_{\tau_j \sim p_\theta(\tau)} \left[\nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \middle| \tau_i \right] \right] \end{aligned} \tag{8}$$

and consider the innermost expectation:

$$\begin{aligned} E_{\tau_j \sim p_\theta(\tau)} \left[\nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \middle| \tau_i \right] &= \int_{\mathbf{s}_{j,t}, \mathbf{a}_{j,t}} p(\mathbf{s}_{j,t}, \mathbf{a}_{j,t}|\pi_\theta, \tau_i) \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \\ &= \int_{\mathbf{s}_{j,t}} p(\mathbf{s}_{j,t}|\pi_\theta, \tau_i) \int_{\mathbf{a}_{j,t}} \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \\ &= \int_{\mathbf{s}_{j,t}} p(\mathbf{s}_{j,t}|\pi_\theta, \tau_i) \int_{\mathbf{a}_{j,t}} \nabla_\theta \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \\ &= \int_{\mathbf{s}_{j,t}} p(\mathbf{s}_{j,t}|\pi_\theta, \tau_i) \nabla_\theta \int_{\mathbf{a}_{j,t}} \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) = \int_{\mathbf{s}_{j,t}} p(\mathbf{s}_{j,t}|\pi_\theta, \tau_i) \nabla_\theta 1 = 0. \end{aligned} \tag{9}$$

Using (4) for the first term and (5) for the second, we can write the expression (6) as

$$\begin{aligned} \nabla_\theta J(\theta) \approx \sum_{i=1}^{N} u(r(\tau_i)) \Bigg(& w'\left(\frac{i}{N}\right) \frac{1}{N} \sum_{j=1}^{i} \sum_{t=1}^{T_j} \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \\ & + w'\left(\frac{i-1}{N}\right) \frac{1}{N} \sum_{j=i}^{N} \sum_{t=1}^{T_j} \nabla_\theta \log \pi_\theta(\mathbf{a}_{j,t}|\mathbf{s}_{j,t}) \Bigg) \end{aligned} \tag{10}$$

Without introducing a bias, we can use (10) to approximate

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} u(r(\tau_i)) \left(\left(w'\left(\frac{i}{N}\right) + w'\left(\frac{i-1}{N}\right) \right) \sum_{t=1}^{T_i} \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}). \right. \tag{11}$$

Note that the weight coefficients $(w'(\frac{i}{N}) + w'(\frac{i-1}{N}))$ should be normalized over each batch. Also, the expression (11) is equivalent to (6) in expectation, but with substantially reduced variance. It has a clear intuition – trajectories are assigned utilities based on their rewards and their contributions to the gradient are scaled by the derivative of the weight function. Standard variance reduction techniques, some requiring additional assumptions, can be applied to this simplified form. Without further assumption or introduction of bias, a static baseline b can be employed:

$$\nabla_\theta J(\theta) \approx \sum_{i=1}^{N} \left(u(r(\tau_i)) - b \right) \left(w'\left(\frac{i}{N}\right) + w'\left(\frac{i-1}{N}\right) \right) \sum_{t=1}^{T_i} \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \tag{12}$$

The additional term is 0 in expectation as

$$\begin{aligned} & E_{\tau_i \sim p_\theta(\tau)} \left[b \left(w'\left(\frac{i}{n}\right) + w'\left(\frac{i-1}{n}\right) \right) \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \right] \\ &= b \left(w'\left(\frac{i}{n}\right) + w'\left(\frac{i-1}{n}\right) \right) \int_{\mathbf{s}_{i,t}, \mathbf{a}_{i,t}} p(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}|\pi_\theta) \left[\nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \right] \\ &= b \left(w'\left(\frac{i}{n}\right) + w'\left(\frac{i-1}{n}\right) \right) \int_{\mathbf{s}_{i,t}} p(\mathbf{s}_{i,t}|\pi_\theta) \int_{\mathbf{a}_{i,t}} \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \\ &= b \left(w'\left(\frac{i}{n}\right) + w'\left(\frac{i-1}{n}\right) \right) \int_{\mathbf{s}_{i,t}} p(\mathbf{s}_{i,t}|\pi_\theta) \nabla_\theta \int_{\mathbf{a}_{i,t}} \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) = 0. \end{aligned} \tag{13}$$

While it is straightforward to choose b to minimize variance, we find little benefit to that choice over the mean of u. The latter is easier to compute, so we evaluate it in our variance reduction experiments.

Further variance reduction may be enacted if we assume that utility may be allocated on a per-step basis. In this case, per-step utilities are computed as the difference between what the full-episode utility would be if the episode were to end at a given time step and what it would have been had the episode ended at the previous time step. While not always applicable, this assumption has the benefit of modeling the temporal allocation of rewards. With it, the variance of (11) may be further reduced through the incorporation of reward-to-go and a state-dependent baseline $V_\phi(\mathbf{s})$:

$$\nabla_\theta J(\theta) \approx \sum_{i=1}^{N} \left(w'\left(\frac{i}{N}\right) + w'\left(\frac{i-1}{N}\right)\right) \sum_{t=1}^{T_i} \nabla_\theta \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \sum_{t'=t}^{T_i} \left(u(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'}) - V_\phi(\mathbf{s}_{i,t'})\right) \tag{14}$$

Here $u(\mathbf{s}_{i,t'}, \mathbf{a}_{i,t'})$ is the per-step utility. The value function $V_\phi(\mathbf{s})$ is parameterized by ϕ and trained via regression to minimize

$$\mathcal{L}(\phi) = \sum_{i,t} \left|\left| V_\phi(\mathbf{s}_{i,t}) - \sum_{t'=t}^{T_i} u(\mathbf{s}_{i,t'}) \right|\right|^2. \tag{15}$$

A standard argument, similar to the approach taken in (Achiam, 2018), can be used to show that the incorporation of reward-to-go does not change the expected value of (11). The addition of a state-dependent baseline also does not introduce a bias, as can be seen by substituting $b \to V_\phi(\mathbf{s}_{i,t})$ in (13) and the fact that $V_\phi(\mathbf{s}_{i,t})$ may also be pulled out of the innermost integral.

Finally, discount factors, bootstrapping, and trust regions may be used to provide additional variance reduction, as they are in conventional on-policy learning. These measures do bias the policy gradient estimate, but typically lead to more sample-efficient learning. In our experiments, we evaluate the use of generalized advantage estimation (GAE; (Schulman et al., 2016)) based on the utility-to-go as well as the clipping-based trust regions of Proximal Policy Optimization (PPO; (Schulman et al., 2017b)). Incorporating these in the policy gradient yields

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(w'\left(\frac{i}{N}\right) + w'\left(\frac{i-1}{N}\right)\right) \sum_{t=1}^{T_i} \nabla_\theta L_{\text{clip}}\Big(\log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}), A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})\Big), \tag{16}$$

where $A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t})$ is the standard GAE except with per-step utilities in place of rewards. Trust regions are implemented similarly to PPO, pessimistically clipping policy updates to be within a multiplicative factor of $1 \pm \epsilon$ of the existing policy:

$$\begin{aligned} L_{\text{clip}} = \min \Bigg(& \log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}), \\ & \log \left(\text{clip}\left(\frac{\pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})}{\pi_{\theta_{\text{old}}}(\mathbf{a}_{i,t}|\mathbf{s}_{i,t})}, 1 \pm \epsilon \right) \pi_{\theta_{\text{old}}}(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}) \right) A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}) \Bigg). \end{aligned} \tag{17}$$

They are used to perform multiple policy updates using the same batch of data, providing learning that is no longer strictly on-policy but that can be significantly more sample efficient. When following this route, we apply the same early stopping mechanism, based on Kullback-Leibler divergence (D_{KL}) between old and new policies, as was used in Ray et al. (2019).

3.4 Learning Algorithm

The above sample-based estimate of the policy gradient may be used to train agents to maximize distributional objectives of the form (2). The resulting method, Cumulative Probability Proximal Policy Optimization (C3PO), is given in Algorithm 1 and mirrors standard on-policy learning.

Algorithm 1 Cumulative Probability Proximal Policy Optimization (C3PO)

Require: Policy: initial parameters θ_0, learning rate α_θ, updates per batch M_θ
Require: Value: initial parameters ϕ_0, learning rate α_ϕ, updates per batch M_ϕ
Require: Early stopping threshold $D_{\text{KL, stop}}$
  for $k = 1, 2, \ldots$ **do**
    Collect set of episodes $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi(\theta_k)$ in the environment
    Compute per-step utilities $u(t)$ and discounted utilities to go $\hat{u}(t)$
    Fit value function by regression to $\hat{u}(t)$:
    for $m = 1, \ldots M_\phi$ **do**
      $\phi \leftarrow \phi + \alpha_\phi \nabla_\phi \sum_{i,t} \left|\left| V_\phi(\mathbf{s}_{i,t}) - \sum_{t'=t}^{T_i} u(\mathbf{s}_{i,t'}) \right|\right|^2$
    end for
    Update utility-based advantage estimates $A_u^\pi(\mathbf{s}, \mathbf{a})$, using new $V_\phi(\mathbf{s})$
    Compute weight coefficients based on ordered episode outcomes and normalize
    Update policy, using KL-based early stopping:
    for $m = 1, \ldots M_\theta$ **do**
      if $D_{\text{KL}}(\pi_\theta || \pi_{\theta_{\text{old}}}) < D_{\text{KL, stop}}$ **then**
        $\theta \leftarrow \theta + \alpha_\theta \frac{1}{N} \sum_{i=1}^{N} (w'(\frac{i}{N}) + w'(\frac{i-1}{N})) \sum_{t=1}^{T_i} \nabla_\theta L_{\text{clip}}(\log \pi_\theta(\mathbf{a}_{i,t}|\mathbf{s}_{i,t}), A_u^\pi(\mathbf{s}_{i,t}, \mathbf{a}_{i,t}))$
      else
        break
      end if
    end for
  end for

Beyond the utility and weight components, Algorithm 1 differs from conventional methods in the requirement to collect full episodes of data in each batch. This requirement can be removed if outcomes can be defined over partial rather than full episodes, an assumption that is often viable. For instance, while out of scope for this work, our approach could be applied to the Atari suite (Bellemare et al., 2013) by considering the outcomes of fixed-length windows and restructuring Algorithm 1 to mimic minibatch PPO (Schulman et al., 2017b).

4 EXPERIMENTS

To evaluate our approach, we sought both to establish that it can effectively optimize different distributional objectives and to explore the impact of using different objectives on agent behavior. We found the OpenAI Safety Gym (Ray et al., 2019) to be suitable for these purposes. Safety Gym is a configurable suite of continuous, multidimensional control tasks wherein different types of robots must navigate through obstacles with different dynamics to perform different tasks. The locations of the goals and obstacles are highly randomized, forcing the agent to learn a generalized navigation strategy. Safety Gym logs adverse events but does not incorporate them into the reward function. As our method relies solely on the training signal from the reward function, we assigned each logged adverse event a fixed, negative reward contribution of -0.025. To highlight distributional differences, we focused on the publicly available, obstacle-rich level 2 environments.[2] Avoiding the longer compute time of the"Doggo" robot, we evaluated the"Point" and "Car" robots on each task ("Goal", "Button", and "Push").

In all experiments, we evaluated five random seeds and matched the hyperparameters used in the baselines accompanying Safety Gym (Ray et al., 2019) as closely as possible. The neural networks used to model both policy and value were multilayer perceptrons (MLPs), with two hidden layers of 256 units each and tanh activations. As in Ray et al. (2019), the policy network outputs the mean values of a multivariate gaussian. The control variances were optimized but independent of state. The full complement of variance reduction measures were used throughout; see Appendix A.1 for experimental justification.

[2]In Safety Gym, the default environments have three levels (0, 1, 2); obstacle density increases with level.

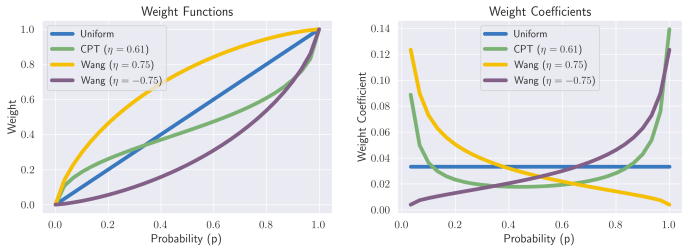

Figure 1: Weight functions and their resulting coefficients in the CDF-based policy gradient (11). In these plots, outcomes increase in quality from left to right.

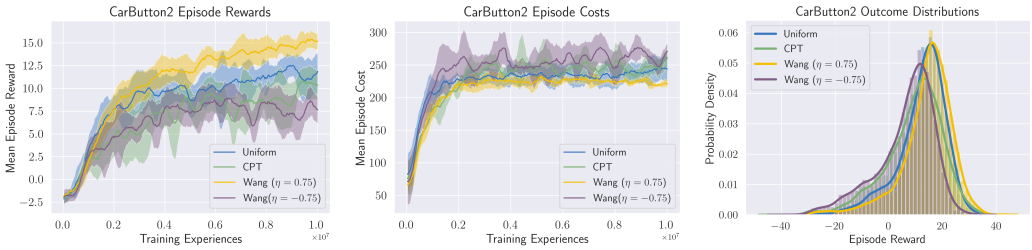

Figure 2: Impact of different distributional objectives in one environment (CarButton2). The shading in the first 2 plots (and subsequent learning curves) reflects the standard deviation associated with running over 5 random seeds. Left: Net reward (positive reward minus penalty) throughout learning. Middle: Average number of cost events per episode during training (lower is better). Right: Agent outcome distribution in testing (i.e., with sampling turned off).

.

4.1 Differing Objectives

Agent performance was explored under four different distributional objectives. In addition to expected reward (the identity weight function) and CPT value (whose form is given in Appendix A.1), we optimized for cautious ($\eta = 0.75$) and aggressive ($\eta = -0.75$) versions of the distortion risk measure proposed in Wang (2000). This measure is defined as $w(p) = \Phi(\Phi^{-1}(p) + \eta)$, where Φ and Φ^{-1} are the standard normal cumulative distribution function and its inverse. While we found this form to be convenient, the "Pow" metric in Dabney et al. (2018a) or any other set of similarly shaped w curves should achieve a similar effect. Also note that, in this set of experiments, the CPT reference point was taken to be the mean episode reward of the current batch in order to match the tendency of humans to change their standards over time. The four weight functions and their resulting coefficients in (11) are shown in Figure 1.

Plots of the total rewards (including penalties), average cost events per episode, and outcome distribution in testing are shown for one environment in Figure 2 and two additional environments in Appendix A.2. The trends were fairly consistent over the three environments evaluated in this manner. While no explicit effort was made to handle cost (agents were given only the sum of positive rewards and penalties), the cautious and aggressive weightings consistently accumulated relatively low and high costs, respectively. The cautious ($\eta = 0.75$) agent typically also generated the highest positive and total rewards at the end of a 10 million step training run.

To generate the histograms in Figures 2, 5, 8, and 12 as well as the numbers in Table 1, the trained agents were deployed on a set of 5000 test episodes – 1000 for each of the 5 networks learned using different random seeds in training. Sampling was turned off, allowing the agents to choose their perceived optimal action at each time step. In this context, the benefit of emphasizing the lower part of the outcome distribution (i.e., cautious weighting) became clear. The effect was more pronounced than may be expected from the first panel of Figure 2, in part because the methods that emphasize poor outcomes tended to maintain higher policy entropy (Appendix A.2).

	PointButton2				CarGoal2				CarButton2			
	Mean	**Std**	**Q=0.5**	**Q=.05**	**Mean**	**Std**	**Q=0.5**	**Q=.05**	**Mean**	**Std**	**Q=0.5**	**Q=.05**
Uniform	22.5	7.1	22.8	10.7	18.1	6.6	18.0	7.4	12.4	9.1	14.0	-6.1
CPT Value	16.5	7.3	16.3	4.9	13.9	7.7	14.6	-0.5	9.5	10.7	11.2	-11.0
Wang (-0.75)	17.5	6.1	17.6	7.6	13.5	6.9	14.0	0.9	6.4	10.7	8.9	-15.5
Wang (0.5)	23.3	6.3	23.3	13.1	18.5	6.0	18.8	**8.2**	12.9	9.0	14.4	-5.6
Wang (0.75)	24.2	6.7	24.4	13.7	19.0	6.4	19.3	8.2	**14.3**	9.5	**15.8**	**-4.4**
Wang (1.0)	24.7	6.0	24.9	15.1	**20.3**	6.9	**21.3**	7.2	11.4	11.0	13.8	-12.2
Wang (1.25)	**25.4**	6.1	**25.4**	**15.9**	17.3	6.7	17.8	5.6	12.7	10.4	14.8	-8.3
Wang (1.50)	23.6	5.9	23.7	14.1	16.6	7.5	17.2	3.1	10.1	12.0	13.3	-17.9
Wang (1.75)	23.4	6.2	23.5	13.5	12.5	8.0	13.0	-0.6	7.3	13.0	11.1	-22.8
PPO	19.0	6.4	18.9	9.1	15.8	6.0	15.8	5.8	8.3	9.5	9.5	-9.8
PPO + Wang(0.75)	20.4	8.4	21.7	2.6	17.7	6.7	18.0	6.3	12.1	9.5	13.6	-6.8

Table 1: Testing statistics for different distributional objectives. $Q = 0.5$ is the median and $Q = 0.05$ refers to the location of the 0.05 quantile. Bold-face represents the best performance for a given environment; in all cases these occur for moderately cautious weightings ($\eta \in [0.75, 1.25]$).

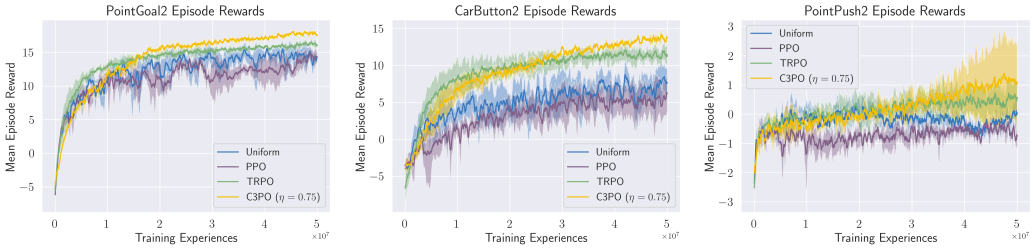

Figure 3: Average episode reward (including penalty) over training for different learning approaches in three different environments. Shading reflects the standard deviation over 5 random seeds.

4.2 Cautious Weightings

To further explore the apparent benefits of cautious weightings in Safety Gym, we trained a series of variably cautious agents by tuning η in the risk-averse weight function proposed by Wang (2000). Histograms of their performance in testing are given in Appendix A.2 and summarized in Table 1. In these environments, agent performance – both in terms of improving the lower end of the distribution and on average – was seen to generally improve with increasing η until the range $\eta \in [0.75, 1.25]$, subsequently degrading. Additional comparisons were made with PPO (Schulman et al., 2017b), which unsurprisingly was found to closely track performance of the "Uniform" agent. We found that naively incorporating cautious weightings into PPO improved its performance (row PPO + Wang(0.75) in Table 1), though not to the level of the full C3PO method with $\eta = 0.75$.

We then pursued a set of longer runs to more carefully study the ability of cautious weightings to improve learning. In these trials, we highlighted the ability of cautious weightings to stabilize learning by doubling the penalty contribution per event to -0.05 and using C3PO with a fixed $\eta = 0.75$ for each environment. Full results for three environments are given in Figures 3, 4, and 5 and for the remaining three environments in Appendix A.3. In addition to PPO, we compared performance with Trust Region Policy Optimization (TRPO; Schulman et al. (2017a)) as configured in Ray et al. (2019). Since this TRPO configuration generally outperformed the PPO configuration in Ray et al. (2019) from which we derived the hyperparameters for C3PO, we would expect C3PO to be at a disadvantage in comparison. However, we found C3PO had the highest average reward (including penalty) in five of the six environments and lowest average cost in five of the six environments.[3] In addition, agents that used the cautious weightings tended to have more stable and repeatable training, as evidenced by the tight distribution of their learning curves. This tightness was found to reflect a lack of negative outlier episodes.

5 Discussion

The analysis above allows for sample-based policy gradient estimates of a general class of distributional objectives. Variance reduction measures were shown to enable efficient optimization based on these estimates (Appendix A.1). However, it was not seen to be the case that a given distributional ob-

[3]The lone exception in the cost category was the CarGoal2 environment, where the cautious agent chose to accept more penalties related to velocity in an effort to reach more goals in the allotted episode duration.

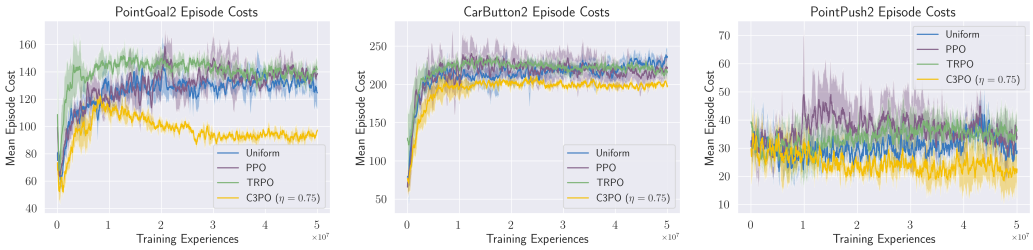

Figure 4: Average number of penalty events per episode over training for different learning approaches in three different environments. Shading reflects the spread over 5 random seeds.

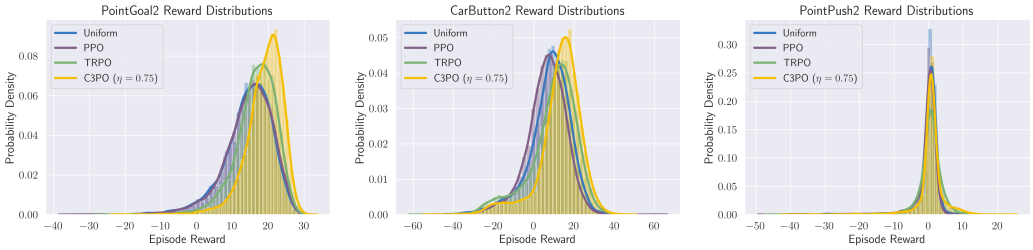

Figure 5: Episode reward (including penalty) distributions for long runs in all six Safety Gym environments in testing (i.e., with sampling turned off).

jective could be most effectively optimized directly. Instead, the best results were generally obtained through moderate emphasis of negative training outcomes (Table 1).

To understand this behavior, consider the interplay of optimization and exploration in the training of cautious and aggressive agents. Cautious weightings continually emphasize the lower part of the outcome distribution, pushing that part of the distribution upward and adjusting behavior the most where it is most necessary. Once an emphasized part of the state space is adequately addressed, a different part of the state space takes its place. In this way, increased attention is always given to the scenarios that most require it. Policy entropy remains high because of the emphasis on problematic situations, ensuring adequate exploration. This trend continues with increasing η, until the point where the agent begins to ignore the higher-quality training outcomes too much. Conversely, aggressive weightings continually emphasize the best outcomes of the distribution. When an already strong outcome is given increased attention, it is likely to stay at the top. Hence agents trained with aggressive weightings tend to become myopic, obsessing over a fraction of the state space while neglecting the rest of it. Ironically, they tend to explore inadequately and fail to attain better top-end performance than more cautious weightings.

Given the consistent performance gains observed with cautious weightings here and the intuition behind them, we propose that they may provide a broadly applicable means for improving the stability and performance of on-policy learners. This should be particularly true when significant penalty terms are present in the reward formulation. In many cases, the application of cautious weightings may be more palatable for adjusting cost tolerances than reward shaping and higher performing than constrained approaches (e.g. Achiam et al. (2017); Ray et al. (2019)). While not an exact match, they are related in spirit to Prioritized Experience Replay (Schaul et al., 2016) for off-policy learning.

6 Conclusions

In this work, we derived a sampling-based estimate of the policy gradient for a broad class of CDF-based distributional objectives. In evaluating agents trained to maximize different objectives, we found that agents that moderately emphasize improvement of poor training outcomes consistently outperformed others. We suggest that the use of such cautious weightings in on-policy learning will often lead to more stable and higher performing learning, particularly when significant penalty terms are present in the reward formulation. Our work provides a foundation for further exploration of risk-sensitive policy gradient methods for complex real-world applications.

REFERENCES

Josh Achiam. Openai spinning up: Proof for don't let the past distract you. http://spinningup.openai.com/en/latest/spinningup/extra_pg_proof1.html, 2018.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 22–31, 2017.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. arXiv:1606.06565, 2016.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, jun 2013.

Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70, pp. 449–458, 2017.

Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. Risk-sensitive and robust decision-making: A CVaR optimization approach. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 1522–1530, 2015.

Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pp. 1096–1105, 2018a.

Will Dabney, Mark Rowland, Marc G. Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pp. 2892–2901, 2018b.

Jingliang Duan, Yang Guan, Shengbo Eben Li, Yangang Ren, Qi Sun, and Bo Cheng. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2021.

Javier García and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(42):1437–1480, 2015.

Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–291, 1979.

Dickson H. Leavens. Diversification of investments. *Trusts and Estates*, 80:469–473, 1945.

Xiaoteng Ma, Li Xia, Zhengyuan Zhou, Jun Yang, and Qianchuan Zhao. DSAC: Distributional soft actor critic for risk-sensitive reinforcement learning. arXiv:2004.14547, June 2020.

German I. Parisi, Ronald Kemker, Jose L. Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, May 2019.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

L.A. Prashanth, Cheng Jie, Michael C. Fu, Steven I. Marcus, and Csaba Szepesvári. Cumulative prospect theory meets reinforcement learning: Prediction and control. In *Proceedings of the 33nd International Conference on Machine Learning (ICML)*, pp. 1406–1415, 2016.

John W. Pratt. Risk aversion in the small and in the large. *Econometrica*, 32:122–136, 1964.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. https://cdn.openai.com/safexp-short.pdf, 2019.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. arXiv:1511.05952, 2016.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv:1707.06347, 2017b.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NeurIPS)*, pp. 1468–1476, 2015.

Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and Uncertainty*, 5(4):297–323, 1992.

Shaun S. Wang. A class of distortion operators for pricing financial and insurance risks. *The Journal of Risk and Insurance*, 67(1):15, March 2000.

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

Congbin Wu and Yuanlie Lin. Minimizing risk models in Markov decision processes with policies depending on target values. *Journal of Mathematical Analysis and Applications*, 231:47–67, 1999.

Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Mingyan Liu, Duane Boning, and Cho-Jui Hsieh. Robust deep reinforcement learning against adversarial perturbations on state observations. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, volume 33, pp. 21024–21037, 2020.

Junyu Zhang, Amrit Singh Bedi, Mengdi Wang, and Alec Koppel. Cautious reinforcement learning via distributional risk in the dual domain. *IEEE Journal on Selected Areas in Information Theory*, 2(2):611–626, 2021.

Han Zhong, Ethan X. Fang, Zhuoran Yang, and Zhaoran Wang. Risk-sensitive deep RL: Variance-constrained actor-critic provably finds globally optimal policy. arXiv:2012.14098, 2020.

A Appendix

A.1 Variance Reduction Measures

To gauge the impact of the variance reduction techniques outlined in Section 3.3, we evaluated their performance in maximizing the value function of Cumulative Prospect Theory (Tversky & Kahneman, 1992). As mentioned in Section 3, this function has two integrals of the form (2):

$$\begin{aligned} J(\theta) = &-\int_{-\infty}^{\infty} u^{-}(r(\tau))\frac{d}{dr(\tau)}\Big(w^{-}(P_\theta(r(\tau)))\Big)dr(\tau) \\ &+\int_{-\infty}^{\infty} u^{+}(r(\tau))\frac{d}{dr(\tau)}\Big(-w^{+}(1-P_\theta(r(\tau)))\Big)dr(\tau) \end{aligned} \tag{18}$$

In Tversky & Kahneman (1992), the utility functions are computed relative to a reference point and reflect the tendency of humans to be more risk-averse in the presence of gains than in the presence

of losses. The weight functions $\{w^+, w^-\}$ model our inclination to emphasize the best and worst possible outcomes in our decision-making.

More specifically, in these experiments we used the piecewise utility functions $u^+(r) = H(r - r_0)(r - r_0)^\sigma$ and $u^- = \lambda H(r_0 - r)(r_0 - r)^\sigma$ with static reference $r_0 = 10$, $\sigma = 0.88$, and $\lambda = 2.25$. The weight function $w(p) = \frac{p^\eta}{(p^\eta + (1-p)^\eta)^{\frac{1}{\eta}}}$ was used, where $\eta = 0.61$ for $r < r_0$ and $\eta = 0.69$ for $r \geq r_0$. Four methods were evaluated, incorporating progressive amounts of variance reduction:

- **Base**: CDF-Based policy gradient with a static baseline (12)
- **UTG**: Base with utility-to-go and a neural network baseline (14)
- **GAE**: UTG with generalized advantage estimation ((16) without clipping)
- **TR**: GAE with trust regions ((16) with clipping (17))

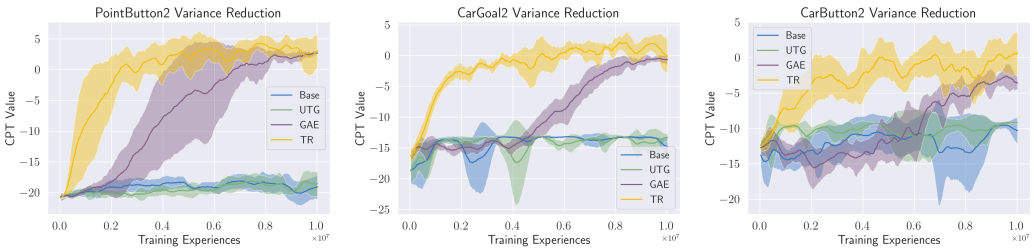

Figure 6: Impact of variance reduction measures on optimization of the CPT value function. Here "Base" refers to the CDF-based policy gradient (12), "UTG" adds utility-to-go and a neural network baseline 14, "GAE" incorporates generalized advantage estimation, and "TR" implements trust regions via clipping. Shading represents the variation over five random seeds.

As shown in Figure 6, the incorporation of these techniques increased the efficiency of the CPT value optimization significantly. Consequently, we used the full complement (TR) in all other experiments.

A.2 Differing Objectives: Additional Results

Below we include results for all environments for the experiments described in Section 4.1.

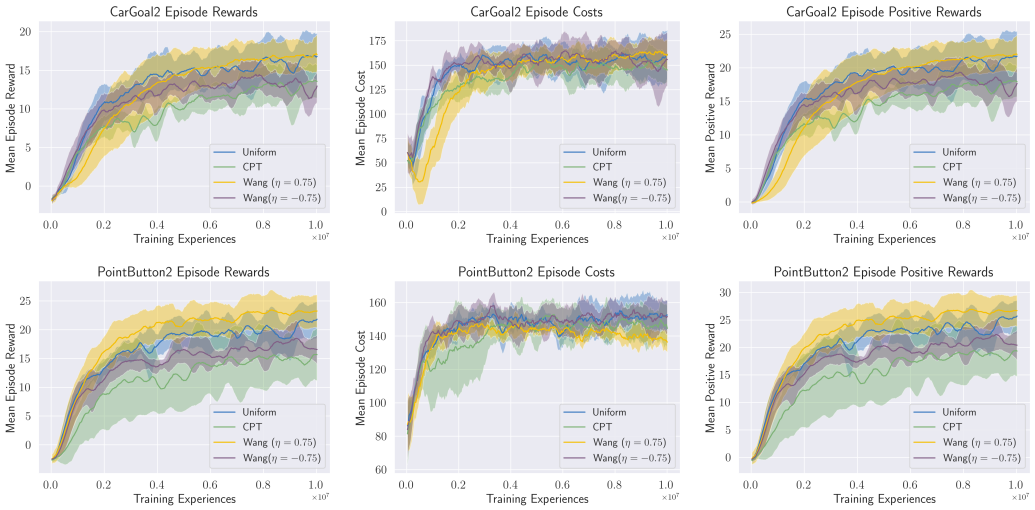

Figure 7: Example learning curves for different distributional objectives. The plot on the left is net reward (positive reward contributions minus penalty), the middle is average cost events per episode during training (lower is better), and the plot on the right is average positive reward accumulated per episode. Shading reflects the observed variation over 5 random seeds.

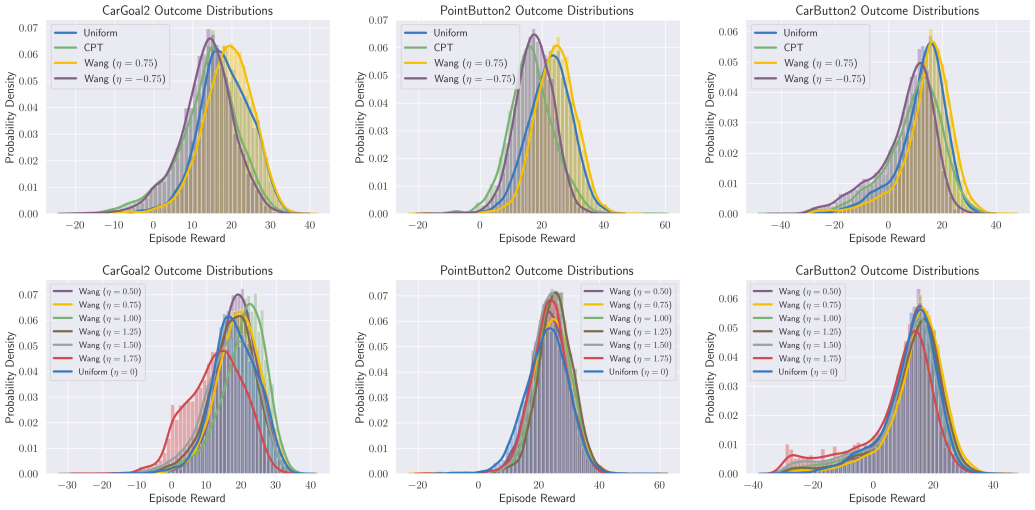

Figure 8: Agent outcome distributions across objectives, with each row corresponding to a different environment. The plots in the top row reflect the initial experiment over different distributional objectives, while the plots in the bottom row correspond to trials run over increasingly cautious (η increasing) agents.

In addition, we note the trend of policy entropies with different distributional objectives. In general, more cautious weightings maintain higher entropy for longer than more aggressive weightings. Note that these plots represent an upper bound because they do not account for action clipping by the environment; see Ray et al. (2019) for details.

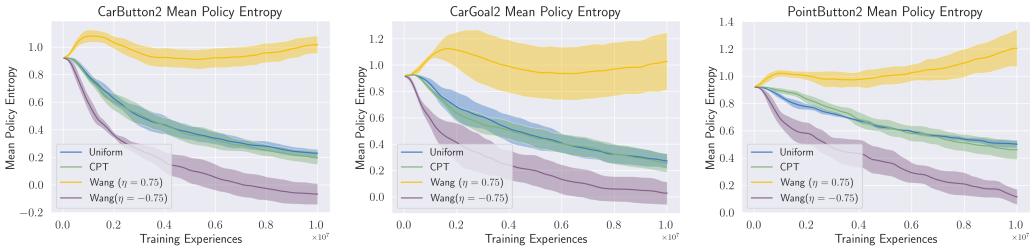

Figure 9: Policy entropy progression during training for three environments. Shading reflects the observed variation over 5 random seeds.

A.3 Long Runs with Cautious Weightings: Additional Results

Below are plots of average episode reward and average number of episode cost events throughout training for the remainder of the environments on which we conducted long runs (Section 4.2). Also included are histograms of testing performance for those runs.

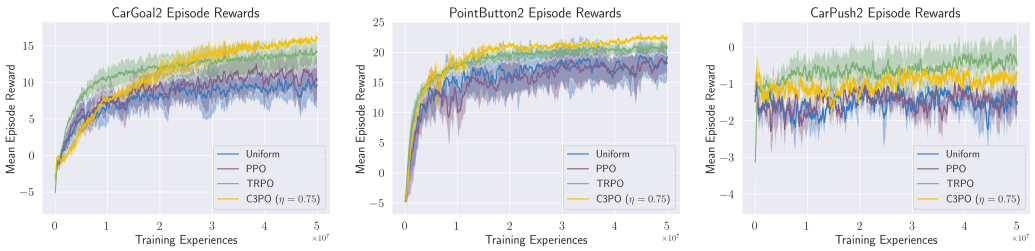

Figure 10: Average episode reward (including penalty) over training for different learning approaches in three different environments. Shading reflects the spread over 5 random seeds.

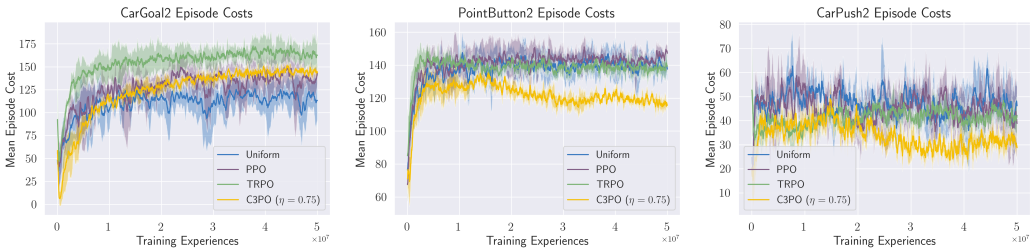

Figure 11: Average number of penalty events per episode over training for different learning approaches in three different environments. Shading reflects the spread over 5 random seeds.

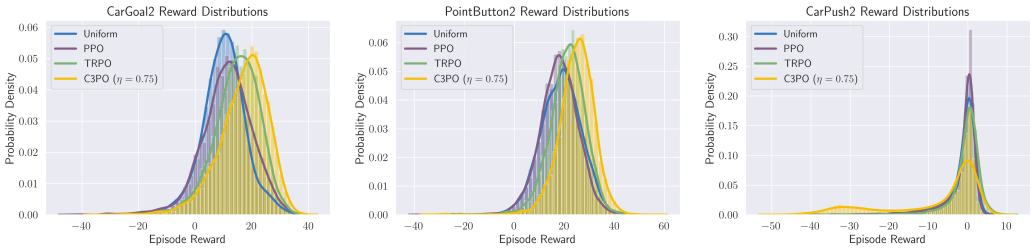

Figure 12: Episode reward (including penalty) distributions for long runs in all six Safety Gym environments in testing (i.e., with sampling turned off). In five of the six environments, C3PO with $\eta = 0.75$ provides tangible benefit. A smaller η is likely required to improve performance on CarPush2.

A.4 Supplementary Materials

The code used to produce these results is included in our Supplementary Materials.