Learning to Be Cautious

Montaser Mohammedalamen
University of Alberta; Amii
Edmonton, Canada
mohmmeda@ualberta.ca

Dustin Morrill
SonyAI
Edmonton, Canada
dustin.morrill@sony.com

Alexander Sieusahai
University of Alberta; Amii
Edmonton, Canada
asieusah@ualberta.ca

Yash Satsangi
Tilburg University
Tilburg, Netherlands
yashziz@gmail.com

Michael Bowling
University of Alberta; Amii
Edmonton, Canada
mbowling@ualberta.ca

ABSTRACT

A key challenge in the field of reinforcement learning is to develop agents that behave cautiously in novel situations. It is generally impossible to anticipate all situations that an autonomous system may face or what behavior would best avoid bad outcomes. An agent that could learn to be cautious would overcome this challenge by discovering for itself when and how to behave cautiously. In contrast, current approaches typically embed task-specific safety information or explicit cautious behaviors into the system, which is error-prone and imposes extra burdens on practitioners. In this paper, we present both a sequence of tasks where cautious behavior becomes increasingly non-obvious, as well as an algorithm to demonstrate that it is possible for a system to *learn* to be cautious. The essential features of our algorithm are that it characterizes reward function uncertainty without task-specific safety information and uses this uncertainty to construct a robust policy. Specifically, we construct robust policies with a k-of-N counterfactual regret minimization (CFR) subroutine given a learned reward function uncertainty represented by a neural network ensemble belief. These policies exhibit caution in each of our tasks without any task-specific safety tuning.

KEYWORDS

Reinforcement Learning, Robustness, Caution

1 INTRODUCTION

One challenge in the field of artificial intelligence (AI) is to design agents that avoid doing harm or being destructive. This is of particular concern in complex, dynamic environments that change over time, where exhaustive training and testing are infeasible. For example, consider a self-driving car that learns to drive through experience. The first time that the car encounters roads covered in snow, two problems arise and compound. Snow makes the environment more hazardous (*e.g.*, tires are more likely to lose traction) and gives the road a new appearance. A natural response is to react by behaving cautiously, *e.g.*, driving more slowly, but current learning algorithms do not develop such an intuition.

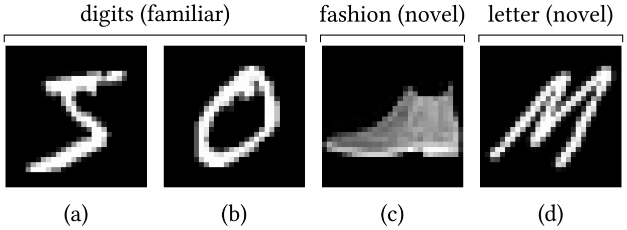


Figure 1: From left to right, image of the numbers five and zero from the MNIST digit dataset, an ankle boot from the MNIST fashion dataset, and a capital "M" from the EMNIST letters dataset. In our motivating example, the two images on the left are similar to the ones you have seen before while the two on the right are novel.

Existing approaches for safety in reinforcement learning (RL) often specify safe behavior via constraints that an agent must not violate [4, 8, 16]. Broadly, this amounts to formulating tasks as a constrained Markov decision processes [2]. A constrained MDP can be solved using reinforcement learning (RL) in a model-based [3, 4] or model-free [1, 8, 31] way. However, this approach requires predefining the safe states that the agent is allowed to visit or the safe actions the agent can take. Alternatively, some approaches design "safety functions" that incentivize pre-defined safe behaviors [37, 38]. Approaches that require an a priori description of safety information about specific scenarios present a scaling problem as it is generally infeasible to enumerate all potentially hazardous situations in a realistic application.

Our goals for this work are (i) to illustrate why it would be useful to design agents that can automatically *learn* cautious behavior, (ii) to describe a series of simple tasks where success requires learned caution, and (iii) to provide an example system that learns to be cautious. We describe a sequence of tasks where cautious behavior is increasingly non-obvious, starting from a one-shot environment (contextual bandit) with an obvious cautious action, leading to one-shot environments with cautious actions that depend on context, and concluding with a gridworld driving environment that requires long-term planning. We show that the k-of-N counterfactual regret minimization (CFR) [6] robust optimization algorithm with a neural network ensemble belief generates policies that identify and adopt cautious behavior in all of these tasks. In the process, we also contribute an extension of k-of-N CFR that is efficient and sound for decision problems without a fixed horizon, like our driving gridworld.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2023), Cruz, Hayes, Wang, Yates (eds.), May 9-10, 2023, Online, https://ala2023.github.io/. 2023.

2 A MOTIVATING EXAMPLE

Consider a decision-making task where you are shown an image and must choose one of eleven actions. The images are hand-drawn digits from MNIST (23, *e.g.*, Figure 1a and b) and you observe a reward of +1 for selecting the action with the index matching the portrayed digit and zero otherwise, except for the eleventh action, which always yields a small reward, +0.25. Now, what do you do when the image is not of a familiar digit but is instead a novel image of a piece of clothing from MNIST fashion (42, *e.g.*, Figure 1c) or a letter from EMNIST letters (11, *e.g.*, Figure 1d)? A natural choice is action eleven, which has always given a reward regardless of the image, while every other action often gave no reward at all. But is this choice common for current AI algorithms?

One obvious approach for choosing the next action is to guess what reward each action will yield with the new image and choose the action with the largest estimate. For example, a nearest-neighbor approach would select a previous image that resembles the new image in some way and use the rewards from the previous image as the reward estimates, effectively extrapolating from familiar to novel images. Considering that the new image looks very different from all the previous ones, an extrapolative approach relies on a questionable premise. Algorithms like this are unlikely to choose action eleven, since its reward was always small and only one of the ten other extrapolations need to look promising for action eleven to be overlooked.

A conventional RL approach like Q-learning [39] or policy gradient [33, 41] with function approximation also employs extrapolative guessing and fails to behave cautiously in this task. After training on MNIST digits, a greedy policy with respect to a single neural network model of the reward function (effectively Q-learning) chooses action eleven less than 2% of the time when presented images from MNIST fashion.

A common approach to achieve caution is to provide prior knowledge about what behaviors are safe. For example, we could designate action eleven as a "safe action" and encourage the agent to choose it when observing a non-digit image or when the agent has no strong preference for any other action. Thomas et al. [35] outlines a general methodology for algorithms of this sort and Kahn et al. [20] provides a more sophisticated example. Embedding prior knowledge about safety into an algorithm would be easy and effective in this particular task but it is problematic as a general approach because safety is highly task-specific and the design burden becomes worse for complicated tasks where safety guarantees would be most useful. In this vein, we present variations on our MNIST task such that cautious behavior becomes increasingly non-obvious.

An alternative to explicitly specifying cautious behavior or safety incentives is risk-sensitive RL. Broadly, these methods characterize an agent's uncertainty about future rewards of different behaviors, and then choose *robust* behaviors, *i.e.*, those that maximize the agent's reward assuming unfavorable conditions (often with a formal risk measure). There are two types of uncertainty that might be present in a decision-making task, (i) *aleatoric* uncertainty that is stochasticity inherent in the environment, *e.g.*, the agent may be uncertain about the the number that a die will show before it is rolled, and (ii) *epistemic* uncertainty that stems from the agent's lack of certainty about the specific environment, *e.g.*, the agent may be uncertain about a die's probability distribution, not just its outcome.

There are various methods for learning policies that are robust to aleatoric uncertainty [7, 10, 34], but since the mapping from images and actions to rewards is deterministic in our MNIST example task, there is no aleatoric uncertainty to be robust to. Consequently, these methods do not behave differently from extrapolative systems in tasks like this.

Alternatively, if the agent is certain about action eleven's reward and less certain about the rewards of the other actions, then a robust policy would choose action eleven, provided the level of uncertainty is great enough. Thus, epistemic uncertainty has the potential to induce caution.[1] In this case, the agent's beliefs are crafted with the domain in mind to achieve the desired behavior in much the same way as the previously discussed prior knowledge approaches. There are many more sophisticated variations on this idea [9, 17, 29, 30, 43], but they share similar downsides as prior knowledge approaches.

Our approach, that we detail for the remainder of the paper, uses robust optimization with a *learned* belief without imposing any task-specific safety information into either component to automatically construct cautious policies. This algorithm learns autonomously to identify and choose cautious behavior that is unique to each task. We evaluate in a sequence of tasks where cautious behavior is increasingly complex. This sequence begins with the MNIST example task described here and escalates to a gridworld driving task that requires sequential decision-making.

3 LEARNING TO BE CAUTIOUS

An agent that interacts with the world and learns from experience will inevitably encounter both familiar and novel situations. We believe that such agents can and should use their previous experience to automatically respond cautiously in novel situations.

Markov Decision Processes. Our tasks use a simplified formulation of the learning-to-be-cautious problem. The agent's world is separated into the familiar and the novel, each represented as a *Markov decision process* (*MDP*).

A finite, discounted MDP, $(\mathcal{S}, \mathcal{A}, p, d_\varnothing, \gamma)$, is a finite set of *states*, $\mathcal{S}$, a finite set of *actions*, $\mathcal{A}$, a Markovian *state transition probability distribution*, $p(\cdot \mid s, a) \in \Delta(\mathcal{S})$ for all states s and actions a (where $\Delta(\mathcal{S})$ is the probability simplex over set $\mathcal{S}$), an initial state distribution $d_\varnothing \in \Delta(\mathcal{S})$, and a discount factor, $\gamma \in [0, 1)$.[2] Feedback evaluating the agent's behavior within an MDP is typically encoded as a scalar *reward function*, $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [-U, U]$, where the magnitude of each reward is bounded by $U \in \mathbb{R}$. This allows us to evaluate a (stationary) policy, π—*i.e.*, an assignment of probability distributions, $\pi(\cdot \mid s) \in \Delta(\mathcal{A})$, to each state s—according to its *γ-discounted expected return.* If $S_0 \sim d_\varnothing$, $A_i \sim \pi(\cdot|S_{i-1})$, and $S_i \sim p(\cdot|S_{i-1}, A_i)$, then the normalized[3] γ-discounted expected

[1] One might be tempted to think that cautious behavior and robustness to epistemic uncertainty are the same. However, as noted, whether robust policies produce cautious behavior is critically dependent on the uncertainty distribution.

[2] We use the $\gamma = 0$ case to address the contextual bandit setting.

[3] As in Kakade [21], we use return functions that are normalized by the effective horizon, $1 - \gamma$, so that returns have the same scale as rewards. This makes optimality approximation errors easier to interpret. See Section 2.2.3 of Kakade [21] for further discussion.

return is

$$v_{\varnothing}(\pi; r) = (1-\gamma)\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(S_i, A_{i+1}, S_{i+1})\right].$$

Furthermore, the algorithms we discuss will make use of

$$\begin{aligned} &q_s(a, \pi; r) \\ &\quad = (1-\gamma)\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i r(S_i, A_{i+1}, S_{i+1}) \,|\, S_0 = s, A_1 = a\right], \end{aligned}$$

which is the (normalized) γ-discounted expected return for taking action a in state s under reward function r and discount factor γ, before following policy π thereafter.

Extrapolation. Since we are primarily interested in examining the agent's *behavior* in novel situations about which they have never received feedback, we do not define a reward function for the novel MDP. We assess the agent's behavior in the novel MDP qualitatively. Here we focus on only reward uncertainty; investigating caution with transition uncertainty needs further investigation both theoretically and practically.

A straightforward approach for the agent to formulate goals for the novel MDP is to extrapolate the familiar reward function. Ordinary RL planning algorithms can then be applied to generate a policy that will perform well if the novel and familiar MDPs are very similar. Extrapolation can be done with conventional regression methods, *e.g.*, we can model the reward function as a neural network and train its parameters by applying an optimization algorithm like stochastic gradient descent to minimize the network's mean squared error. A natural approach, given an extrapolated reward function model, $\hat{r}$, is then to behave according to an optimal policy, *e.g.*, in each state s, set $\pi^{\mathrm{Optim}(\hat{r})}(a \,|\, s) = 1$, where action a is the first action (under an arbitrary ordering) that maximizes $q_s(\cdot, \pi^{\mathrm{Optim}(\hat{r})}; \hat{r})$. This approach will represent a simple non-cautious baseline in our experiments.

Inference. A fundamental problem with extrapolation is that there are typically multiple reward models that match the familiar reward function but differ in novel situations from the novel MDP (*i.e.*, state, action, next state triples not present in the familiar MDP), even within a restricted model class. To address this issue, we can infer a posterior belief (a probability distribution) about which reward models are more reasonable, given a prior belief that describes what it means for a reward model to be "reasonable". Exact Bayesian inference is typically intractable for high dimensional data, but a convenient approximation is to train an ensemble of neural networks, each with unique initialization parameters and trained on independently shuffled familiar examples. Each neural network acts like a sample from a posterior with an implicit prior so that the entire ensemble implicitly characterizes a posterior-like belief. Various previous works (*e.g.*, Heskes et al. [19], Lakshminarayanan et al. [22], Lu and Van Roy [25], Osband et al. [27], Pearce et al. [28], Tibshirani [36]) have used neural networks in similar ways to characterize uncertainty with connections to proper Bayesian inference.

Robust Optimization. An inference approach characterizes the agent's uncertainty about what reward functions are reasonable in the novel MDP given the familiar MDP, but ordinary RL algorithms cannot make use of this information beyond optimizing for a single reward function generated from the belief (*e.g.*, a sample, the expected posterior, or the maximum a posteriori reward function). Robust policy optimization algorithms however, are designed to learn policies that are robust to such uncertainty.

The *k-of-N counterfactual regret minimization* (*k-of-N CFR*) algorithm computes an approximate $\mu_{k\text{-of-}N}$-robust policy, which is a policy that approximately minimizes the k-of-N risk measure, $\mu_{k\text{-of-}N}$ [6]. This Bayesian risk measure is closely related to the classic conditional value at risk (CVaR) measure. By tuning the $k > 0$ and $N \geq k$ parameters, the algorithm designer can set a desired robustness level between worst-case ($k = 1$ and N large) and average-case ($k = N$). As N is increases, $\mu_{k\text{-of-}N}$ approximates the CVaR measure at the k/N percentile. k-of-N CFR works by iteratively sampling N reward functions from a belief and updating the current policy to improve its value under the k-worst reward functions.

As described in Section 2, it is critical to pair robust optimization with an appropriate belief for cautious behavior to emerge. Often this is achieved by manually tailoring the belief to specific aspects of a task, but can we instead use a generic neural network ensemble to induce caution? Consider the belief that such an ensemble would learn from training data where the reward for one action is a constant, as in the example from Section 2. If, as is common, the training procedure has any preference for neural networks with small weights, then all of the last layer weights corresponding to the constant reward action in all of the neural networks will converge toward zero and their bias terms will converge toward the constant. Since all neural networks agree about the reward for this action in all states, the ensemble belief is always nearly certain about the reward of this action. Uncertainty about the rewards for other actions caused by disagreement between neural networks in the ensemble pushes a robust policy into choosing the constant reward action.

The experiments in the next section show that k-of-N CFR under a neural network ensemble belief can effectively learn to be cautious in various tasks.[4] A limitation of k-of-N CFR that we overcome in this work is that it has only been described for fixed horizon MDPs, *i.e.*, those that terminate after a fixed number of decisions. We show how it can be applied in any continuing MDP, but we defer these details to Section 5 in favor of experimental results that first illustrate the utility of learning to be cautious.

4 EXPERIMENTS

We now present a sequence of tasks that require agents to automatically learn cautious behavior. Tasks vary in difficulty from one that requires no sequential reasoning and includes a universal cautious action, to one that requires sequential reasoning and where the return from each action is context dependent, with a natural progression in-between. Experimental design details and hyperparameters for the algorithms tested are provided in Appendix A.

[4] Other algorithms that are robust to epistemic uncertainty, *e.g.*, Chow et al. [9], Ghavamzadeh et al. [17], Petrik and Subramanian [29], Zahavy et al. [43] could potentially be used instead of k-of-N CFR.

Learning to Ask for Help. Our first task is the previously described decision making task with MNIST images. The familiar states are the 60,000 training images in the MNIST digit dataset, where the initial state and each next state is sampled uniformly at random. Ten of the actions correspond to a digit label and a reward of +1 is given when the label matches the image and zero otherwise. The eleventh action can be thought of as an "ask for help" option that always receives a reward of +0.25. All action labels are solely to aid our discussion whereas the agent only observes action indices. The discount factor is zero so the agent's return is simply their reward, making this a contextual bandit task.

The k-of-N CFR procedure iteratively improves an approximately robust policy by evaluating it on N samples from a belief updating the policy according to the k-worst samples. Thus, for T iterations, we need to train NT neural networks. We train 2000 reward models on the familiar MDP so that we can run 100 CFR iterations with a maximum $N = 20$. These models also provide the basis for the Optim($\hat{r}$) baseline, where each neural network in the ensemble represents an extrapolated reward model, $\hat{r}$. We set $N = 20$ $k = 1$ for the most robustness, $N = 10$ $k = 1$ for moderate robustness, and $N = 10$ $k = 5$ for marginal robustness. We represent each k-of-N CFR instance with the last policy generated after 100 iterations.

We construct novel MDPs with 10,000 images from the MNIST fashion [42] test set and 20,800 images from EMNIST letters [11] test set (lower and uppercase). Using the set of images as a set of states, we construct two novel MDPs with two different state distribution schemes representing two evaluation scenarios. The first scenario replicates the dynamics of the familiar MDP in that each image is always sampled uniformly. This describes a task where the agent must come up with a policy that works well on all novel images, without emphasizing the performance given any particular one. Our second scenario uses a point-mass initial state distribution and identity transition distribution. This scenario corresponds to a decision-making task where a single crucial novel state is given instead of a distribution over multiple possible novel states. In this scenario, the impact of robustness is exaggerated because the k-of-N CFR policy trains on the k-worst reward functions specifically targeted to a single state rather than the k-worst averaged across many states. Figure 2a shows the results of both experiments.

In both state distribution regimes, the classification accuracy of all policies, including the most robust k-of-N policies, on the 10,000 images in the MNIST digit test set, ranges from 96% to 99%. The two most robust policies, 1-of-20 and 1-of-10, choose the help action 2% and 1% of the time respectively in the single-image regime, but the rest of the policies across both regimes almost never choose the help action. This uniformity in behavior is caused by the fact that our neural networks effectively generalize to all MNIST digit test images, making the ensemble belief accurate and confident on these images.

The "all fashion images" scenario replicates our motivating example and shows that the help action is utilized more on the fashion images by k-of-N policies as k is decreased (*i.e.*, with more risk aversion), up to 29% of the time for 1-of-20. The Optim($\hat{r}$) baseline is the least likely to use the help action on each novel dataset, and this propensity does not change substantially with the dataset. Decreasing the k/N ratio causes the k-of-N policies to increase the help action frequency on the letter images from 3% to 6%.

In the single-image regime, 1-of-20 selects the help action 89% of the time on the fashion images and 68% on the letter images—46 and 69 times more often, respectively, than the Optim($\hat{r}$) baseline. And when 1-of-20 does not select the help action with the letter images, it does so for letters that resemble digits, *e.g.*, o, s, i, l, j, and z resemble 0, 5, 1, and 2. See Appendix A for more details, including confusion matrices of selected actions.

Discovering Non-Obvious Cautious Actions. Our next task is to discover non-obvious cautious actions where the value of each action is input-dependent. This time, there are only ten actions and the reward for action indexed as $a \in \{0, \ldots, 9\}$ is $(a + 1)$ if a is the correct label for a given digit image or $-{}^{(a+2)}/_{9}$ otherwise. The reward for a correct classification scales with the action index, but so does the cost of misclassification. This reward function also ensures that always choosing action zero has the same expected value as guessing the digit at uniform random assuming a balanced distribution of digit images. Thus, policies that choose lower index actions are more cautious.

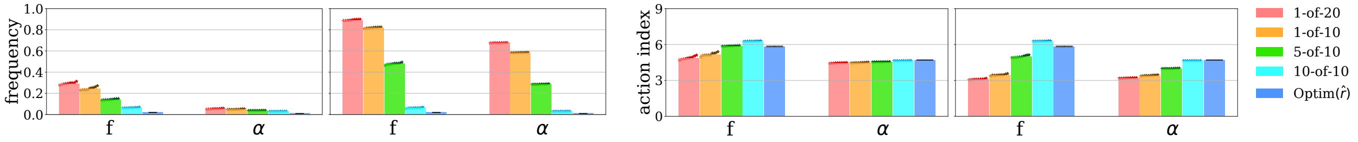
Again, we evaluate our approach in two regimes, one where the set of novel states is an MNIST test set and another where evaluation is done on each of these images individually. Figure 2b shows the average action index chosen by each algorithm in each novel environment.

Again, all polices correctly label nearly all test digit images. In both regimes, evaluating on the fashion images, we see that 1-of-20 and 1-of-10 systematically choose smaller actions at lower indices than non-robust algorithms, and 5-of-10 is intermediate between 1-of-20 and 10-of-10 along this metric. The differences are smaller on the letter images in the all-images regime, likely due to many similarities between letter and digit images, but the ordering of methods according to robustness is preserved in both regimes.

Ask for Help Only When it is Available. In the previous scenarios, cautious actions could be identified without taking features of the input into account. Here we modify the previous task where lower index actions are generally more cautious to have an extra action, as in the "learning to ask for help" task, but the value of this action changes depending on an input feature. This feature is a signal that help is available, in which case the "ask for help" action receives a reward of $+{}^{1}/_{20}$. The "ask for help" action is therefore better than any incorrect classification and worse than correctly classifying even the least valuable digit (zero) if there is help available. If help is unavailable, the "ask for help" action is the worst action as it always receives a reward of $-{}^{11}/_{9}$. Figure 3 show the results for each method in the all-images and single-image regimes (left and right, respectively).

Evaluating on fashion images, we see that the robust methods with $k < N$ select the help action much more than the non-robust methods when help is available. When help is unavailable, these methods never select the help action and instead choose actions with smaller indices. The average action index decreases much more when help becomes available because policies switch from choosing actions with high indices to choosing the help action.

How Caution Depends on the Extent of Training Data. Do the k-of-N policies really *learn* to be cautious? Here we investigate



(a) The average frequency of the help action in (left) the all-images regime and (right) the single-image regime.

(b) Average action index chosen in (left) the all-images regime and (right) the single-image regime.

Figure 2: Results for the "learning to ask for help" and "discovering non-obvious cautious actions" tasks in each novel environment ("f" for fashion and "α" for letters).

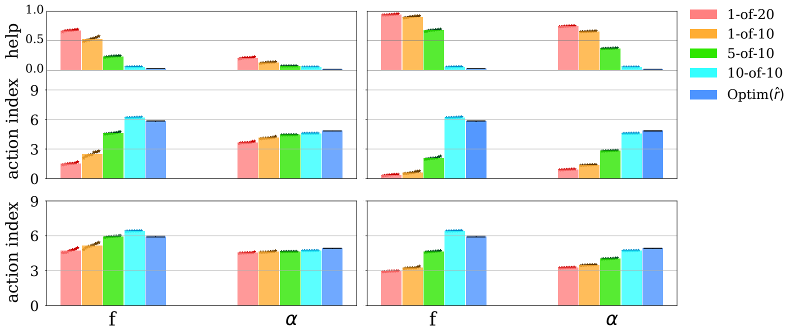


Figure 3: Average action index and help action frequency chosen by each method in each novel environment in the "ask for help only when it is available" task. (top row) Help is available, (bottom row) help is unavailable, (left column) the all-images regime, (right column) the single-image regime. All methods essentially never choose the help action when help is unavailable.

how cautious algorithms behave depending on the extent of training data. We repeat the "learning to ask for help" task except that rewards are perturbed by white noise (with standard deviation 0.1) once before neural network training, and the neural network training data varies between 1%, 10%, and 100% of the full digit dataset. Noise is added so that it takes more than a single training example to learn that the expected reward of the help action is constant across training images. Results are shown in Figure 4.

When reward models are trained with 1% of the digit images, we observe that decreasing k to increase robustness does not induce caution. Effectively, the neural network ensemble belief has not seen enough data to infer that the "ask for help" action yields a small but consistent reward. Increasing the training set size to 10%, the correlation between robustness and caution returns and is even stronger than when the full digit dataset is used for training. This shows that caution requires enough training data for the agent to accurately infer the training reward function, and once achieved, the robust agents find cautious behavior.

Driving Gridworld. For a sequential decision making task, we introduce a gridworld driving environment (see Figure 5 for an example frame) in the spirit of the AI safety gridworlds [24]. A state is a five column image, where the first and second columns represent a two-lane road, the outer two columns represent a ditch, and the last column represents a speedometer. The agent's car is on the bottom row of the image and the world shifts down as the car drives forward. The height of the image represents how far ahead the driver can see. An obstacle randomly appears on the new parts of the road revealed when the car moves forward. To keep the number of states in the gridworld small, only one obstacle can be present on both the left and right halves of the gridworld at a time, and we use a vision range of two. The car's speed limit is the vision range plus one so that they can "overdrive" their vision by one unit.

The agent has five actions: change lane left, change lane right, accelerate, brake, and cruise. Accelerate and brake increases or decreases the car's speed by one unit, respectively. The car always moves according to its current speed, so the impact of accelerating or braking on the distance the car travels is only felt in later time steps. The car changes lanes one space at a time and changing lanes requires momentum so the car must not be stopped and it travels forward by one fewer space than it would otherwise. The car's speed and lane does not change if the agent chooses to cruise.

The agent's goal is to drive as far from their starting location as possible. As there is no fixed destination, the task is naturally represented as a continuing MDP. The agent receives a reward of +1 for each space it moves forward, −2 for each ditch space it moves over, and −2 times the current speed of the car when it drives over an obstacle.

We investigate how our algorithm reacts to novel situations by restricting obstacles to the two ditches in the familiar MDP and allowing them to appear on the road in the novel MDP. We build our ensemble belief by training 2000 neural networks to mimic the familiar MDP's reward function and each k-of-N CFR instance (where $k \in \{1, 2, 4, 5, 10, 20\}$ and $N = 20$) is represented by the last policy generated after 100 iterations.

Figure 5 shows that the more robust policies drive slower, and drive over obstacles both less frequently and at slower speeds in

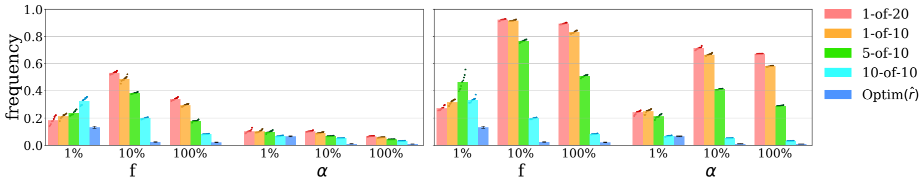


Figure 4: The average frequency of the help action in each novel environment on the "learning to ask for help" task with perturbed rewards, where reward models are trained on only 1%, 10%, or 100% of the digit dataset. (left) The all-images regime, (right) the single-image regime.

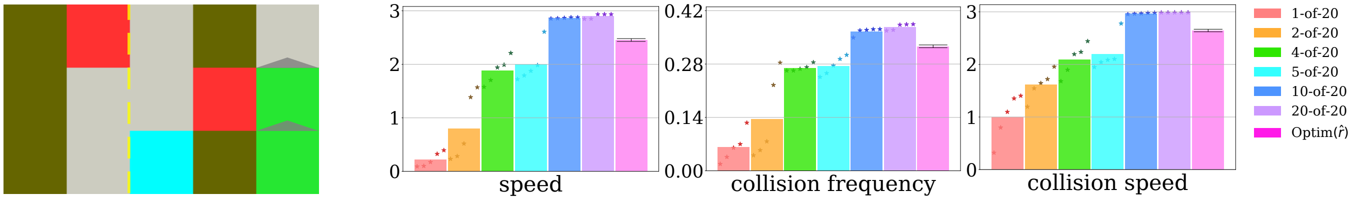


Figure 5: Left: a frame from the driving gridworld environment. The cyan square is the car, the red squares are obstacles, and the rightmost column is the car's speedometer. Right: normalized γ-discounted safety statistics for each algorithm in the driving gridworld.

the novel MDP, reflecting intuitively cautious driving behavior. The non-robust policies in contrast almost always drive at full speed.

Why do we see this difference? Since obstacles are never observed on the road in the familiar MDP, there is no clear signal that driving over these obstacles will cause a bad outcome. There is a clear signal however that driving fast on the road yields larger rewards, so the non-robust policies optimize their behavior around this signal, which is reflected in the belief's average reward function. The robust policies instead take the belief's uncertainty about what could happen when the car drives over an obstacle on the road into account. Since there are some reward functions in the ensemble belief that generalize collisions in the ditch with collisions on the road, the agent learns to avoid collisions altogether in the novel MDP.

5 k-OF-N CFR FOR CONTINUING MDPS

We now provide theoretical support for the efficient application of CFR and k-of-N CFR to discounted continuing MDPs, like the driving gridworld, with reward uncertainty.

CFR is a conceptually simple algorithm: compute the expected return of choosing each action given the current policy from each *decision point* (*e.g.*, a state, state–time step pair, or state history sequence) weighted according to the policies of other agents and transition probabilities; update the policy at each decision point to optimize these *counterfactual values*; and iterate. CFR operates on decision points rather than states directly because extra context can be used to get more reward if the horizon is known in advance or if other agents can influence the transition probabilities. CFR updates its policy according to a *no-regret* learning rule that ensures the average counterfactual value from each decision point approaches the average counterfactual value from any single action taken in that decision point. In practice, these learning rules typically cause the policy to take a small step towards the greedy policy with respect to the current counterfactual value on each iteration.

The key property of CFR is that regret is minimized across all decision points jointly even though CFR only directly minimizes regret at each decision in isolation [44]. Chen and Bowling [6] prove as a consequence that if the transition distribution and reward function on each iteration are sampled uniformly from the k-worst of N candidates sampled from a belief, then CFR approximates an optimal policy under the k-of-N robustness measure with respect to that belief.

CFR's procedure makes no restriction on the environment except that there are a finite number of decision points and actions, and the expected return from each decision point exists. However, CFR's regret bound was originally proven in the extensive-form game framework where the decision points are equivalent to *state histories* in a *fixed horizon* MDP. That is, CFR's policy must condition on the entire history of states and all states eventually lead to terminal states that lack actions or outward transitions. Chen and Bowling [6] argue that as long as there is no transition uncertainty (so there may still be reward uncertainty), then it is safe to instead define decision points as state–time step pairs.

Since the transition model never changes, it is actually safe and convenient to implement k-of-N CFR with reward uncertainty only with expected returns rather than counterfactual values. Once this change is made, it becomes apparent that we could re-derive the regret and optimality bound for CFR and k-of-N CFR, respectively, in fixed horizon MDPs with reward uncertainty using the undiscounted half of Kakade [21]'s performance difference lemma (Lemma 5.2.1). The other half of the performance difference lemma provides an analogous statement for stationary policies in discounted continuing MDPs. Even-Dar et al. [15] uses an average reward version of the performance difference lemma to analyze what is essentially CFR for the average reward objective in continuing MDPs. Our analysis instead considers the discounted return objective and directly addresses the regret and robustness of k-of-N CFR in specific.[5]

Let $v_s(\pi; r) = \mathbb{E}_{A \sim \pi(\cdot|s)}[q_s(A, \pi; r)]$ denote the (normalized) γ-discounted expected return of policy π from state s and let $\rho_s(a, \pi; r) = q_s(a, \pi; r) - v_s(\pi; r)$ be the (normalized) expected *advantage* of choosing action a over π in s. Define

$$d_s : s'; \pi \mapsto (1-\gamma)\mathbb{E}\left[\sum_{i=0}^{\infty} \gamma^i \mathbb{1}\{S_i = s'\} \mid S_0 = s\right],$$

where $A_i \sim \pi(\cdot|S_{i-1})$ and $S_i \sim p(\cdot|S_{i-1}, A_i)$ for $i \geq 1$, to be the γ-discounted future state distribution induced by π from initial state s. Kakade [21]'s performance difference lemma for this setting is:

Lemma 1. *The full regret for using stationary policy π instead of stationary competitor policy π' from state s in MDP $(\mathcal{S}, \mathcal{A}, p, d_\varnothing, \gamma)$ under reward function r is*

$$v_s(\pi'; r) - v_s(\pi; r) = \frac{1}{1-\gamma}\mathbb{E}[\rho_S(A, \pi; r)],$$

where $S \sim d_s(\cdot; \pi')$ and $A \sim \pi'(\cdot|S)$.

From Lemma 1, we derive a new regret and optimality bound for CFR and k-of-N CFR, respectively, in continuing MDPs with reward uncertainty. Given a sequence of reward functions, $(r^t)_{t=1}^T$, CFR produces a sequence of policies, $(\pi^t)_{t=1}^T$, that ensures the cumulative advantage of each action a at each state s grows sublinearly, *i.e.*, $\sum_{t=1}^T \rho_s(a, \pi^t; r^t) \leq C^T \in o(T)$ for bound C^T that depends on the state-local learning algorithm used. For example, it may use *regret matching* [18] instances at each state that learns from $q_s(\cdot, \pi^t; r^t)$ and generates $\pi^t(\cdot \mid s)$ to get a bound of $C^T = 2U\sqrt{|\mathcal{A}|T}$. Combining this with Lemma 1, we arrive at CFR's cumulative full regret bound (all proofs in Appendix B):

Theorem 1. *CFR bounds cumulative full regret with respect to any stationary policy π as*

$$\sum_{t=1}^{T} v_\varnothing(\pi; r^t) - v_\varnothing(\pi^t; r^t) \leq C^T/(1-\gamma).$$

Taking into account Monte Carlo reward function sampling, k-of-N CFR thus inherits the following regret bound:

Theorem 2. *With probability $1-p$, $p > 0$, the full regret of k-of-N CFR with respect to any stationary policy, π, is upper bounded by*

$$\frac{C^T}{1-\gamma} + 4U\sqrt{2T \log 1/p}.$$

Finally, our theoretical inquiry culminates in the following optimality approximation bound for k-of-N CFR policies:

Theorem 3. *With probability $1-p$, $p > 0$, the best policy in the sequence of policies generated by k-of-N CFR, $(\pi^t)_{t=1}^T$, is an ε^T-approximation to a $\mu_{k\text{-of-}N}$-robust policy where*

$$\varepsilon^T = \frac{C^T}{(1-\gamma)T} + 4U\sqrt{\frac{2\log 1/p}{T}}$$

and with probability at least $(1-p)(1-q)$, $q > 0$, a randomly sampled policy from this sequence is an ε^T/q-approximation to a $\mu_{k\text{-of-}N}$-robust policy.

Thus, as long as a no-regret algorithm is deployed at each state in k-of-N CFR so that C^T grows sublinearly with T, the sequence of policies generated by k-of-N CFR converges toward a $\mu_{k\text{-of-}N}$-robust policy with high probability.

6 CONCLUSIONS

Our proof of concept algorithm based on a neural network ensemble and k-of-N CFR shows that algorithms can learn to be cautious. Our testbeds are simple, they capture key aspects of AI safety, and they facilitate experimental comparisons. Our hope is that algorithms that learn to be cautious can improve the safety of, and our confidence in, deployed AI systems. However, this level of automated safety is meant to enhance, *not replace*, human judgement and safety planning.

Transition certainty is a strong assumption that will need to be relaxed for most practical applications of these ideas. The increased difficulty of computing robust policies or even minimizing regret with transition uncertainty is discussed by Chen and Bowling [6] and Even-Dar et al. [15]. It appears an algorithm must search through policies that condition on the entire state history to be sound, which makes policies infeasibly complex in typical environments. Both theoretical and experimental work is required to overcome this hurdle.

Critical limitations of our k-of-N CFR implementation are that it is tabular and requires exact policy evaluation on each iteration to determine the worst-k reward functions. CFR has been used with function approximation [5, 13, 14, 26, 32, 40] and approximate worst-case policy evaluation [12], so applying these enhancements can allow our approach to scale to more complicated environments.

ACKNOWLEDGMENTS

Computation provided by WestGrid and Compute Canada. Dustin Morrill and Michael Bowling are supported by the Alberta Machine Intelligence Institute (Amii), CIFAR, and NSERC. Thanks to John D. Martin for constructive feedback on an early draft of this paper.

REFERENCES

[1] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. 2017. Constrained policy optimization. In *International Conference on Machine Learning*. 22–31.

[2] Eitan Altman. 1999. *Constrained Markov decision processes*. Vol. 7. CRC Press.

[3] Anil Aswani, Humberto Gonzalez, S Shankar Sastry, and Claire Tomlin. 2013. Provably safe and robust learning-based model predictive control. *Automatica* 49, 5 (2013), 1216–1226.

[4] Felix Berkenkamp, Matteo Turchetta, Angela P Schoellig, and Andreas Krause. 2017. Safe model-based reinforcement learning with stability guarantees. In *Neural Information Processing Systems*.

[5] Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. 2019. Deep counterfactual regret minimization. In *International Conference on Machine Learning*. 793–802.

[5] The similarity between CFR for the discounted return objective which we analyze here and Even-Dar et al. [15] analysis for the average reward objective also implies that our analysis of k-of-N CFR algorithm could easily be repeated for the average reward objective, achieving similar regret and robustness guarantees.

[6] Katherine Chen and Michael Bowling. 2012. Tractable objectives for robust policy optimization. In *Neural Information Processing Systems*. 2069–2077.

[7] Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. 2017. Risk-constrained reinforcement learning with percentile risk criteria. *The Journal of Machine Learning Research* 18, 1 (2017), 6070–6120.

[8] Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. 2018. A lyapunov-based approach to safe reinforcement learning. In *Neural Information Processing Systems*. 8103–8112.

[9] Yinlam Chow, Aviv Tamar, Shie Mannor, and Marco Pavone. 2015. Risk-sensitive and robust decision-making: a CVaR optimization approach. *Neural Information Processing Systems* 28 (2015), 1522–1530.

[10] William R Clements, Bastien Van Delft, Benoît-Marie Robaglia, Reda Bahi Slaoui, and Sébastien Toth. 2019. Estimating risk and uncertainty in deep reinforcement learning. In *Workshop on Uncertainty and Robustness in Deep Learning at International Conference on Machine Learning*.

[11] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. 2017. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks*. 2921–2926.

[12] Trevor Davis. 2015. *Using Response Functions for Strategy Training and Evaluation*. Master's thesis. University of Alberta.

[13] Ryan D'Orazio. 2020. *Regret minimization with function approximation in extensive-form games*. Master's thesis. University of Alberta.

[14] Ryan D'Orazio, Dustin Morrill, James R. Wright, and Michael Bowling. 2020. Alternative function approximation parameterizations for solving games: An analysis of f-regression counterfactual regret minimization. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

[15] Eyal Even-Dar, Sham M Kakade, and Yishay Mansour. 2005. Experts in a Markov decision process. In *Neural Information Processing Systems*. 401–408.

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

[17] Mohammad Ghavamzadeh, Marek Petrik, and Yinlam Chow. 2016. Safe policy improvement by minimizing robust baseline regret. *Neural Information Processing Systems* 29 (2016), 2298–2306.

[18] S. Hart and A. Mas-Colell. 2000. A Simple Adaptive Procedure Leading to Correlated Equilibrium. *Econometrica* 68, 5 (2000), 1127–1150.

[19] TM Heskes, WAJJ Wiegerinck, and HJ Kappen. 1997. Practical confidence and prediction intervals for prediction tasks. *Progress in Neural Processing* (1997), 128–135.

[20] Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. 2017. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182* (2017).

[21] Sham Machandranath Kakade. 2003. *On the sample complexity of reinforcement learning*. Ph.D. Dissertation. UCL (University College London).

[22] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Neural Information Processing Systems*. 6405–6416.

[23] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[24] Jan Leike, Miljan Martic, Victoria Krakovna, Pedro A Ortega, Tom Everitt, Andrew Lefrancq, Laurent Orseau, and Shane Legg. 2017. AI safety gridworlds. *arXiv preprint arXiv:1711.09883* (2017).

[25] Xiuyuan Lu and Benjamin Van Roy. 2017. Ensemble sampling. In *Neural Information Processing Systems*. 3260–3268.

[26] Dustin Morrill. 2016. *Using regret estimation to solve games compactly*. Master's thesis. University of Alberta.

[27] Ian Osband, Benjamin Van Roy, Daniel J Russo, and Zheng Wen. 2019. Deep exploration via randomized value functions. *Journal of Machine Learning Research* 20, 124 (2019), 1–62.

[28] Tim Pearce, Mohamed Zaki, and Andy Neely. 2018. Bayesian neural network ensembles. In *Workshop on Bayesian Deep Learning, Neural Information Processing Systems*.

[29] Marek Petrik and Dharmashankar Subramanian. 2014. RAAM: The benefits of robustness in approximating aggregated MDPs in reinforcement learning. *Advances in Neural Information Processing Systems* 27 (2014), 1979–1987.

[30] Marc Rigter, Bruno Lacerda, and Nick Hawes. 2021. Risk-averse Bayes-adaptive reinforcement learning. *arXiv preprint arXiv:2102.05762* (2021).

[31] Krishnan Srinivasan, Benjamin Eysenbach, Sehoon Ha, Jie Tan, and Chelsea Finn. 2020. Learning to be safe: Deep RL with a safety critic. *arXiv preprint arXiv:2010.14603* (2020).

[32] Eric Steinberger, Adam Lerer, and Noam Brown. 2020. DREAM: Deep regret minimization with advantage baselines and model-free learning. *arXiv preprint arXiv:2006.10410* (2020).

[33] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Neural Information Processing Systems*. 1057–1063.

[34] Yichuan Charlie Tang, Jian Zhang, and Ruslan Salakhutdinov. 2020. Worst cases policy gradients. In *Conference on Robot Learning*. 1078–1093.

[35] Philip S Thomas, Bruno Castro da Silva, Andrew G Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. 2019. Preventing undesirable behavior of intelligent machines. *Science* 366, 6468 (2019), 999–1004.

[36] Robert Tibshirani. 1996. A comparison of some error estimates for neural network models. *Neural Computation* 8, 1 (1996), 152–163.

[37] Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. 2016. Safe exploration in finite Markov decision processes with Gaussian processes. *Neural Information Processing Systems* (2016).

[38] Akifumi Wachi and Yanan Sui. 2020. Safe reinforcement learning in constrained Markov decision processes. In *International Conference on Machine Learning*. 9797–9806.

[39] Christopher John Cornish Hellaby Watkins. 1989. *Learning from delayed rewards*. Ph.D. Dissertation. King's College, Cambridge.

[40] Kevin Waugh, Dustin Morrill, J. Andrew Bagnell, and Michael Bowling. 2015. Solving games with functional regret estimation. In *AAAI Conference on Artificial Intelligence*.

[41] Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning* 8, 3 (01 May 1992), 229–256. https://doi.org/10.1007/BF00992696

[42] Han Xiao, Kashif Rasul, and Roland Vollgraf. 2017. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* (2017). MIT license.

[43] Tom Zahavy, Andre Barreto, Daniel J Mankowitz, Shaobo Hou, Brendan O'Donoghue, Iurii Kemaev, and Satinder Singh. 2020. Discovering a set of policies for the worst case reward. In *International Conference on Learning Representations*.

[44] Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. 2008. Regret Minimization in Games with Incomplete Information. In *Neural Information Processing Systems*, Vol. 20. 1729–1736.