Multi-Agent Reinforcement Learning with Prospect Theory

Dominic Danis, Parker Parmacek, David Dunajsky, Bhaskar Ramasubramanian*†

Abstract

Recent advances in cyber and cyber-physical systems have informed the development of scalable and efficient algorithms for these systems to learn behaviors when operating in uncertain and unknown environments. When such systems share their operating environments with human users, such as in autonomous driving, it is important to be able to learn behaviors for each entity in the environment that will (i) recognize presence of other entities, and (ii) be aligned with preferences of one or more human users in the environment. While multi-agent reinforcement learning (MARL) provides a modeling, design, and analysis paradigm for (i), there remains a gap in the development of strategies to solve (ii). In this paper, we aim to bridge this gap through the design, analysis, and evaluation of MARL algorithms that recognize preferences of human users. We use cumulative prospect theory (CPT) to model multiple human traits such as a tendency to view gains and losses differently, and to evaluate outcomes relative to a reference point. We define a CPT-based value function, and learn agent policies as a consequence of optimizing this value function. To this end, we develop MA-CPT-Q, a multi-agent CPT-based Q-learning algorithm, and establish its convergence. We adapt this algorithm to a setting where any agent can call upon 'more experienced' agents to aid its own learning process, and propose MA-CPT-Q-WS, a multi-agent CPT-based Q-learning algorithm with weight sharing. We evaluate both algorithms in an environment where agents have to reach a target state while avoiding collisions with obstacles and with other agents. Our results show that agent behaviors after learning policies when following MA-CPT-Q and MA-CPT-Q-WS are better aligned with that of human users who might be placed in the same environment.

1 Introduction

Complex cyber and cyber-physical systems (CPS), including autonomous cars and drones, rely on the seamless integration of computation and physical components. Multiple CPS operating in the same environment generate large amounts of data; moreover, any single entity (agent) may not have complete information about the parameters and behaviors of other entities in the environment. In such a case, the CPS might depend on machine learning algorithms for decision making. Over the horizon of its operation, each agent will have to solve a sequential decision making problem in order to learn behaviors to optimize its performance to meet desired goals (e.g., safety, reachability) in uncertain and unknown environments. Reinforcement learning (RL) [1] and optimal control [2] are two paradigms that have been commonly leveraged to solve the sequential decision making under uncertainty problem.

Decision making can be *risk-neutral*, where an agent learns strategies (sequence of actions) to maximize an expected reward, when the reward is provided by the environment [1]. Risk-neutral methods have been successfully implemented in multiple domains, including robotics, autonomous vehicles, games, and mobile networks [3–9]. However, these may not be adequate in situations that involve human decision makers.

With recent advances in the development of autonomous driving algorithms and human-robot collaboration strategies [10–12], it will be common for a CPS to share an environment with human users in the near future. Humans have been shown to demonstrate emotional and cognitive biases; specifically, humans might have a different perception of both, the utility and the probabilistic outcome as a consequence of their decisions [13]. In these cases, it will be important for autonomous CPS to demonstrate *risk-sensitive* decision making abilities in a way that will be aligned with the preferences of one or more human users.

The above challenges are exacerbated when there are multiple learning-based CPS sharing an environment. In such a setting, each agent interacts not only with the environment, but also with other agents [14, 15]. As a consequence, as agent' behaviors evolve, the environment has been proven to become non-stationary from the perspective of any single agent [14–16]. We direct the reader to the survey [16] for an overview of the state-of-the-art in multi-agent RL (MARL). The literature examining incorporation of risk-sensitivity in

*The authors are with Electrical and Computer Engineering, Western Washington University, Bellingham, WA 98225, USA. {ramasub}@wwu.edu

†This work was supported by the US National Science Foundation via grant CNS-2153136.

MARL is limited. Recently, the authors of [17, 18] developed a framework to learn risk-sensitive policies in cooperative MARL using the conditional value at risk (CVaR) [19–22]. The CVaR is the average value of a random variable conditioned on the event that the variable takes sufficiently large values. In comparison, in this paper, we use empirical models from the social sciences to represent human preferences, and design MARL algorithms to allow agents to learn behaviors that are aligned with the preferences of one or more human users who might share their environment.

Empirical models from economics and social science have shown that humans derive utility relative to a reference point [23], and tend to be more sensitive to losses than gains [24]. As an illustration, human drivers on the road (i) are more sensitive to changes in speed, than its absolute value; (ii) show greater dislike when being passed (loss) than passing another car (gain); (iii) overestimate small probabilities of engine failures and underestimate large probabilities of running out of gas. Cumulative prospect theory (CPT), introduced in [25], incorporates these properties via non-linear transformations of utility functions and probability distributions.

This paper extends a recent line of research on integrating insights from prospect theory with reinforcement learning [26–29]. Single-agent RL algorithms for CPT-based decision making in unknown environments were developed in [26, 27]. This work established theoretical guarantees on convergence and demonstrated that behaviors of agents using CPT-based policies closely mimicked those of a human user in the same environment. A desire for human users to keep decision-making and its outcomes hidden from external, and possibly adversarial parties was incorporated into a CPT-based RL framework in [28], where we provided guarantees on the privacy of an algorithm while ensuring that agents learned behaviors which were aligned with the preferences of human users. The evolution of opinions in a population of interacting agents that had cognitive biases was analyzed in [29].

In this paper, we develop a framework for prospect-theoretic decision making in multi-agent reinforcement learning. We develop an iterative procedure to enable agents to learn policies and demonstrate its convergence. We also develop a modified variant of the above procedure where an agent will be able to use information from 'more experienced' agents using a weight-sharing strategy [30]. We carry out experiments to show that agent behaviors learned when following a CPT-based algorithm mimic those of a human user more closely than when following a baseline MARL algorithm. We make the following contributions:

- We define the CPT-Q value of a state-action pair and develop a procedure to mitigate the challenge of the curse of dimensionality.
- We design the MA-CPT-Q algorithm for multi-agent CPT-based RL, and show its convergence.
- We design the MA-CPT-Q-WS algorithm to enable an agent to glean information from other 'more experienced' agents.
- We evaluate the MA-CPT-Q and MA-CPT-WS algorithms in an environment where a target state has to be reached while avoiding obstacles and collisions with other agents.

The remainder of this paper is organized as follows: Sec. 2 provides background on RL and CPT. Sec. 3 gives a procedure to mitigate the challenge of the curse of dimensionality in MARL. Sec. 4 and Sec. 5 present our CPT-based MARL Algorithms. Sec. 6 presents experimental results, and Sec. 7 concludes the paper.

2 Preliminaries

2.1 Partially Observable Stochastic Games We model the environment of the agents as a partially observable stochastic game (POSG) [31].

Definition 2.1. *A POSG is a tuple* $\mathcal{G} = (I, S, A, O, T, r^i)$ *where* I *is a set of agents,* S *is a finite set of states,* $A = A_1 \times \cdots \times A_n$ *is a finite set of actions available to each agent* $i \in I$*,* $O = O_1 \times \cdots \times O_n$ *is a finite set of observations for each agent,* T *is a transition function that encodes the probability that taking a joint action* $a = (a_1, \ldots, a_n)$ *in state* s *transits to a state* s' *while emitting a joint observation* $o = (o_1, \ldots, o_n)$*,* $r^i : S \times A \to \mathbb{R}$ *is the reward received by agent* i *when taking action* a *in state* s*.*

A (stochastic) policy is a map π from states to (probability distributions over) actions. Each agent aims to learn a policy π^i to maximize its expected accumulated discounted reward, $\mathbb{E}_{\pi^i}[\sum_t \gamma^t r^i(s_t, a_t)]$, where $\gamma \in (0, 1)$ is a discount factor.

2.2 Risk Measures and Prospect Theory For a set $\mathcal{Y}$ of random variables on Ω, a *risk measure* or *risk metric* is a map $\rho : \mathcal{Y} \to \mathbb{R}$ [32]. The risk metric that we adopt in this paper is informed from cumulative prospect theory [25]. Empirical models from the social sciences have shown that human players or operators demonstrate a preference to play safe with gains and take risks with losses [13]. Further, they tend to overestimate or *deflate* high probability events, and underestimate or *inflate* low probability events. Cumulative prospect theory (CPT) is a risk measure

that has been empirically shown to capture human attitude to risk [25,33]. This risk metric uses two *utility functions* u^+ and u^-, corresponding to gains and losses, and *weight functions* w^+ and w^- that reflect the fact that values seen by a human subject are nonlinear in the underlying probabilities [24].

When Y is a discrete random variable with finite support, let p_i denote the probability of incurring a gain or loss y_i, where $y_1 \leq \cdots \leq y_l \leq 0 \leq y_{l+1} \leq \ldots y_K$, for $i = 1, 2, \ldots, K$. Define $F_k := \sum_{i=1}^{k} p_i$ for $k \leq l$ and $F_k := \sum_{i=k}^{K} p_i$ for $k > l$.

Definition 2.2. *The CPT-value of Y is defined as:*

$$\begin{aligned}&\rho_{cpt}(Y)\\ &:= \Big(\sum_{i=l+1}^{K-1} u^+(y_i)\big(w^+(F_i) - w^+(F_{i+1})\big) + u^+(y_K)w^+(p_K) \Big)\\ &\quad - \Big(u^-(y_1)w^-(p_1) + \sum_{i=2}^{l} u^-(y_i)\big(w^-(F_i) - w^-(F_{i-1})\big) \Big),\end{aligned}$$

where utility functions $u^+, u^- : \mathbb{R} \to \mathbb{R}_{\geq 0}$ are continuous, have bounded first moment such that $u^+(x) = 0$ for all $x \leq 0$, and monotonically non-decreasing otherwise, and $u^-(x) = 0$ for all $x \geq 0$, and monotonically non-increasing otherwise. The probability weighting functions $w^+, w^- : [0,1] \to [0,1]$ are Lipschitz continuous and non-decreasing, and satisfy $w^+(0) = w^-(0) = 0$ and $w^+(1) = w^-(1) = 1$.

The function u^+ is typically concave on gains, while $-u^-$ is typically convex on losses [25]. The distortion of extremely low and extremely high probability events by humans can be represented by a weight function that takes an *inverted S-shape-* i.e., it is concave for small probabilities, and convex for large probabilities (e.g., $w(\kappa) = \exp(-(-\ln \kappa)^\eta), 0 < \eta < 1$) [25,34]. The CPT-value generalizes other risk metrics for appropriate choices of weighting functions. For example, when w^+, w^- are identity functions, and $u^+(x) = x, x \geq 0$, $u^-(x) = -x, x \leq 0$, we obtain $\rho_{cpt}(Y) = \mathbb{E}[Y]$.

2.3 Q-Learning and Nash Equilibria Q-Learning [35] is an RL algorithm to learn an optimal policy π in a finite-state, finite-action environment with provable convergence guarantees. The value of a state s following policy π is denoted $V^\pi(s) := r(s,a) + \gamma \sum_{s'} P(s'|\pi)V^\pi(s')$. The optimal policy is obtained by choosing at each step, the action a that maximizes $V^\pi(s)$. The Q-learning algorithm begins with an initial value of $Q(s,a)$ for each state-action pair, and iteratively updates these values following interactions of the agent with its environment, as $Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha[r(s,a) + \gamma \max'_a Q(s',a')]$, where α is a parameter that trades-off the previous Q-value and the value obtained following an action taken by the agent. When there are $n > 1$ agents, one way to assess performance is by examining convergence of their Q-values to an *equilibrium*. A Nash equilibrium is a point from where there is no incentive for any agent to deviate.

Definition 2.3. *[36] A Nash equilibrium is a tuple of policies $(\pi^1_*, \ldots, \pi^n_*)$, where $\pi^i : S \to A_i$, such that for all states $s \in S$, all agents i, and all possible choices of policies π^i, $V^i(s, \pi^1_*, \ldots, \pi^n_*) \geq V^i(s, \pi^1_*, \ldots, \pi^i, \pi^{i+1}_*, \ldots, \pi^n_*)$.*

In this paper, we restrict our focus to stationary policies. A result from [37] establishes that there always exists an equilibrium in stationary policies.

Theorem 2.1. *[36,37] Every n-player discounted game has at least one Nash equilibrium in stationary policies.*

3 One-Step-Ahead States and CPT Estimation

A challenge that is often encountered (even in finite-state, finite-action setups) in MARL is the curse of dimensionality. As the size of the state space and number of agents increases, memory and computational requirements to learn policies in the environment scales exponentially. To partially mitigate this challenge, we propose *one-step ahead (OSA) state groupings*, an approach inspired from *tile-coding* [1].

3.1 One-Step Ahead State Groupings The OSA procedure is described in Algo. 1 and Fig. 1 shows an illustration. At each step, for each agent, OSA computes the set of possible states s' where the agent will transition into after executing a possible action. OSA also determines states s'' where another agent, if located there, can transition into a state in the set s'. States s' and s'' are represented by blue dots in Fig. 1. Computing OSA states enables determining whether an agent will collide with another agent; consequently, this will inform learning policies to avoid collisions.

Algorithm 1 One Step Ahead Observation

Require: Active agent state s
1: **Initialize** observation, $o = 0 \in \mathbb{R}^{|A|}$
2: Identify OSA states (blue dots in Fig. 1)
3: Determine OSA state groups for each action available to agent (black curves enclosing blue dots in Fig. 1)
4: Check if OSA states in each group are occupied by another agent
5: If group j contains occupied state, label o_j as 1
6: return o

The OSA procedure results in a significant reduction in the computational complexity of the learning

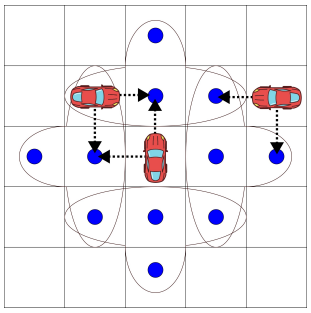

Figure 1: One-step ahead state groupings in a 5×5 grid with the active agent at the center. The blue dots indicate (i) states s' to which the active agent can move by taking an action (*left, right, up, down*), and (ii) states s'' from which another agent can move into one of the states s' in (i) in a single step by taking an action. For each possible action of the active agent, the boundary (black curve) around a region of blue dots indicates the one-step-ahead states for that action.

process. In an $L \times L$ grid with n agents, total amount of memory required to store the locations of each agent is $O(L^{2^n})$. In comparison, using OSA (Algorithm 1), the amount of memory required is reduced to $O(L^2 \times 2^{2^{|A|}})$, which is significantly smaller, since typically, $|A| << L^2$.

3.2 Calculating ρ_{cpt} from samples Algorithm 2 provides a method to obtain multiple samples of a random variable X. These samples are then used to estimate $\rho_{cpt}(X)$. This way to estimate the CPT-value of a random variable was proposed in [33], and was shown to be asymptotically consistent. In order to modify this procedure to the multi-agent setting, we adapt the Nash-Q learning algorithm from [36]. Specifically, we compute the Nash equilibrium for each experience, and assume that each agent plays a Nash equilibrium strategy for all subsequent time steps. The Nash equilibrium (NE) policy $(\pi^1, \ldots, \pi^n)$ for a stage game $Q^i_{cpt}(s')$ at iteration t is computed as:

$$NashQ^i_{cpt}(s') = \pi^1(s') \ldots \pi^n(s') \cdot Q^i_{cpt}(s') \tag{3.1}$$

The $NashQ^i_{cpt}$ value is then used to update Q-values of all agents, which we will describe in the next section.

4 CPT-based MARL Algorithm

Our algorithms for CPT-based MARL are grounded on temporal difference (TD) techniques. TD methods seek to learn value functions using episodes of experience, i.e., a sequence of states, actions, and rewards when following a particular policy. The predicted values at any time-step is updated in a way to bring it closer to the prediction of the same quantity at the next time-step. We adapt a procedure from [33] in order to use TD-methods for CPT-based RL.

Algorithm 2 CPT-Estimation

Require: Global state S, global action A, current policy π, max. samples N_{mx}

Initialize $n = 1$; $X_0 := 0$; $s_* \leftarrow s$
repeat
  Take all actions A, observe $r_i(s, a)$ and next global state S'
  Obtain next observation o' using Algorithm 1
  $X_n := r(s^i, a^i) + \gamma \sum_b \pi(b|\cdot) NashQ^i_{cpt}(b))$
  if $X_n > X_0$ **then**
    $s^i_* \leftarrow s'$
    $X_0 \leftarrow X_n$
  end if
  $n \leftarrow n + 1$
until $n > N_{max}$
Arrange samples $\{X_i\}$ in ascending order: $X_{[1]} \leq X_{[2]} \leq \ldots$
Let:

$$\rho^+_{cpt} := \sum_{i=1}^{N_{mx}} u^+(X_{[i]})(w^+(\frac{N_{mx}+i-1}{N_{mx}}) - w^+(\frac{N_{mx}-i}{N_{mx}}))$$

$$\rho^-_{cpt} := \sum_{i=1}^{N_{mx}} u^-(X_{[i]})(w^-(\frac{i}{N_{mx}}) - w^-(\frac{i-1}{N_{mx}}))$$

$\rho_{cpt}(r^i(s^i, a^i) + \gamma \sum_b \pi(b|\cdot) NashQ^i_{cpt}(b)) := \rho^+_{cpt} - \rho^-_{cpt}$ where $NashQ^i_{cpt}(b)$ is defined in Equation (3.1)
return $\rho_{cpt}(\cdot), s_i*$

We detail the development of the MA-CPT-Q algorithm, a MARL algorithm that adapts the classical Q-learning algorithm [35] to incorporate CPT-based techniques. Since transition probabilities and rewards are not known apriori, agents will have to learn behaviors through repeated interactions with the environment.

Similar to [36], Q-values are defined using the immediate reward that the agent receives, and the discounted sum of future costs, assuming that all agents follow a computed Nash equilibrium for all future steps. Different from [36], we use the CPT-operator from Def. (2.2). The CPT-Q-value of agent i at a state s and joint action a is given by:

$$Q^i_{cpt}(s, a) = \rho_{cpt}(r^i(s, a) + \gamma \sum_{s'} \mathbb{P}(s'|s, a) V^i_{cpt}(s'|\pi^1, \ldots, \pi^n)) \tag{4.2}$$

The MA-CPT-Q algorithm is presented in Algorithm 3. Starting from an initialization of Q-values for all agents, the algorithm updates the Q-values following a TD procedure based on the experiences of agents'

Algorithm 3 MA-CPT-Q Algorithm

Require: Learning rate α; max. episodes T_{max}; discount γ; exploration rate ϵ (decays with number of iterations)

Initialize $Q^i_{cpt}(s,a)$ for all agents, $T=1$
repeat
  For each agent initialize $s \in S$
  repeat
    for all agents **do**
      Obtain o using Algorithm 1
      Select a from π with probability ϵ
    end for
    for all agents **do**
      Obtain $\rho^1_{cpt}(\cdot)...\rho^n_{cpt}(\cdot), s^1_*...s^n_*$ using Algorithm 2
      Obtain $\delta^1...\delta^n$ using $\delta^i = \rho^i_{cpt}(\cdot) - Q^i_{cpt}(s,a)$
      for $j = 1, ..., n$ **do**
        $Q^j_{cpt}(s,a) \leftarrow Q^j_{cpt}(s,a) + \alpha\delta$
      end for
      $s \leftarrow s_*$
    end for
  until s is a terminal state
  $T \leftarrow T+1$
until $T > T_{max}$

interactions with the environment. We have:

$$Q^i_{cpt} \leftarrow Q^i_{cpt} + \alpha\delta^i; \quad \delta^i = \rho_{cpt}(\cdot) - Q^i_{cpt}(s,a) \tag{4.3}$$

In order to establish the convergence of Algorithm 3, we will prove that an operator characterizing updates of the Q-values for the agents will be a contraction.

Definition 4.1. *In an $n-$player POSG, the operator $\mathcal{P}$ is defined for each $i = 1, ..., n$ as: $\mathcal{P}Q^i_{cpt}(s,a) = \rho_{cpt}(r^i(s,a) + \gamma \sum_{s'} \mathbb{P}(s'|s,a)\pi^1(s')\dots\pi^n(s')Q^i_{cpt}(s'))$, where $(\pi^1(s')\dots\pi^n(s'))$ is the NE solution for the stage game $(Q^1_{cpt}(s'), \dots, Q^n_{cpt}(s'))$ from Eqn. (3.1).*

We make the following additional assumptions.

Assumption 4.1. *Each state, observation, and action tuple is visited infinitely often.*

Assumption 4.2. *The learning rate α satisfies $\sum_k \alpha_{k,i} = \infty, \sum_k \alpha^2_{k,i} < \infty$*

We denote the goal equilibrium Q-values in a stage game for the agents as $(Q^1_{cpt} \dots Q^n_{cpt})^*$. From Lemma 10 in [36] (and also [38]), the relationship between the optimal state value function of agent i in state s, $V^i_{cpt}(s)$, and its equilibrium value $(Q^i_{cpt})^*$ is:

$$V^i_{cpt}(s) = \pi^1(s)\dots\pi^n(s)(Q^i_{cpt}(s,a))^*. \tag{4.4}$$

This allows us to present our main result.

Proposition 4.1. *For an $n-$player stochastic game $\mathcal{P}Q^*_{cpt} = Q^*_{cpt}$, where $Q^*_{cpt} = (Q^1_{cpt}, \dots, Q^n_{cpt})^*$. Thus, the operator $\mathcal{P}$ in Def. 4.1 converges to a fixed-point.*

Proof. Establishing that $\mathcal{P}$ is a contraction follows in a similar manner to Lemma 16 in [36], and we omit it due to space constraints. We show that any Nash equilibrium of a stage-game is also a fixed point of $\mathcal{P}$.

From Eqns. (4.2) and (4.4), we have:

$$\begin{aligned} Q^{*i}_{cpt}(s,a) &= \rho_{cpt}(r^i(s,a) \\ &\quad + \gamma \sum_{s'} \mathbb{P}(s'|s,a)V^i_{cpt}(s'|\pi^1, \dots, \pi^n)) \\ &= \rho_{cpt}(r^i(s,a) \\ &\quad + \gamma \sum_{s'} \mathbb{P}(s'|s,a)\pi^1(s'), \dots, \pi^n(s')Q^{*i}_{cpt}(s')) \\ &= \mathcal{P}Q^{*i}. \end{aligned}$$

Since this holds for all agents $i \in \{1, \dots, n\}$, we have $\mathcal{P}Q^*_{cpt} = Q^*_{cpt}$, which completes the proof. □

We refer the reader to [36] for an in-depth coverage of possible corner cases, with the caveat that our learning algorithm works with the CPT-operator $\rho_{cpt}(\cdot)$ rather than expected value $\mathbb{E}(\cdot)$.

5 CPT-based MARL with Weight Sharing

Algorithm 4 MA-CPT-Q-WS Algorithm

Require: Learning rate α; episodes T_{max}; discount γ; exploration rate ϵ

Initialize $Q^i_{cpt}(s,a)$ for all agents i, $T=1$
repeat
  Learn $Q^i_{cpt}(s,a)$ using Lines 2-19 of Algorithm 3
  for all agents i **do**
    for all agents j, $j \neq i$ **do**
      Compute weights: $W^{ij} \leftarrow e^j / \sum_k e_k$
    end for
    Share Q-values:
    $Q^i_{cpt}(s,a) \leftarrow Q^i_{cpt}(s,a) + \sum_j W^{ij} Q^j_{cpt}(s,a)$
  end for
  $T \leftarrow T+1$
until $T > T_{max}$

We present a scheme to model the possibility for an agent to incorporate information from other agents in order to aid its own learning. We adopt the *Normal(Nrm)* expertness criteria proposed in [30], which is computed as an algebraic sum of rewards received from the start of the learning horizon ($t=0$) to the current time-step (t_C). The expertness of agent i is given by $e^i := \sum_{t=1}^{t_C} r^i_t$. The intuition underpinning this is that agents that receive a higher reward are considered to be 'more' expert. Each agent then updates its Q-value using a weighted average of other agents' Q-values as: $Q^i_{cpt} \leftarrow Q^i_{cpt}(s,o,a) + \sum_{j \neq i}(W_{ij} Q^j_{cpt}(s,o,a))$.

The sharing of Q-values enables faster learning, especially for agents who might receive uneven experience.

We describe the CPT-based weight-sharing procedure in Algorithm 4. An agent first learns its Q-values following Algorithm 3. It can then choose to leverage insights gleaned by other agents during the learning process (Line 4 onwards in Algo. 4). Exploring other expert criteria examined in [30] along with providing a rigorous theoretical analysis for CPT-based value functions is an interesting direction for future research.

6 Experimental Evaluation

This section presents an experimental evaluation of the MA-CPT-Q (Algorithm 3) and MA-CPT-Q-WS (Algorithm 4) algorithms. We compare behaviors learned by the agents when optimizing a CPT-based reward signal (ours) with a baseline where agents optimize an expected cost [14]. We also provide a comparison with the weight-sharing strategy using the Nrm criterion [30].

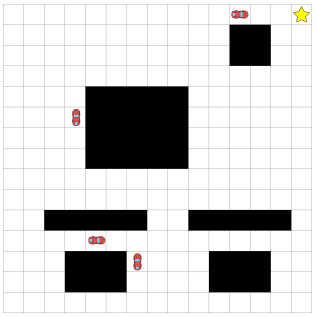

Figure 2: The 15×15 grid environment on which we evaluate Algorithms 3 and 4. We present results for 4 agents that need to learn policies to avoid collisions with obstacles (black boxes) and with each other in order to reach the target state (yellow star at top-right).

6.1 Environment The environment for our experiments is shown in Fig. 2. Agents have to learn behaviors to reach the target state (top right corner) as quickly as possible while avoiding obstacles (black shaded boxes), and collisions with each other. At each step, agents may choose an action from the set $\{up, down, left, right\}$. A *neighboring* state is defined as any state that shares a boundary with the current state of an agent, and we denote the number of neighboring states at the current state by N_{ns}. For each action that the agent takes at a state, the agent can transit with probability 0.9 to the intended next state, and with probability $0.1/(N_{ns}-1)$ to some neighboring state. However, this transition probability is not known to the agent.

We compare results from our algorithms with Independent Q-learning [14] and Q-learning using a weight sharing strategy [30]. We use the same values of learning rate and training iterations. For the CPT-based methods we use utility and weighting functions proposed in [25] that were empirically shown to model various cognitive biases exhibited by humans. We had also used these utility and weighting functions in [26, 28]:

$$u^+(x) = |x|^{0.88}; \quad \omega^+(\kappa) = \frac{\kappa^{0.61}}{(\kappa^{0.61} + (1-\kappa)^{0.61})^{\frac{1}{0.61}}};$$
$$u^-(x) = |x|^{0.88}; \quad \omega^-(\kappa) = \frac{\kappa^{0.69}}{(\kappa^{0.69} + (1-\kappa)^{0.69})^{\frac{1}{0.69}}}.$$

6.2 Results Fig. 3 compares the total number of collisions (with either an obstacle or another agent) when using each algorithm to learn policies. The results reported are over 100 sample paths generated after policies have been learned using the respective algorithms. We observe that agents that have learned policies following our MA-CPT-Q (blue bars) and MA-CPT-Q-WS (green bars) algorithms consistently report fewer collisions than agents learning policies using independent Q-learning [14] (orange bars) or Q-learning with weight-sharing [30] (red bars).

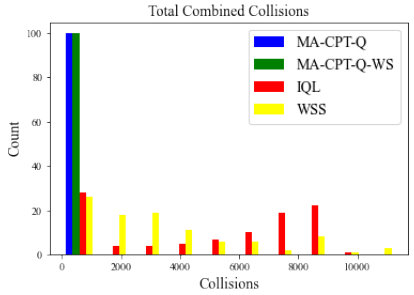

Figure 3: Comparison of total number collisions between pairs of agents and with obstacles when following policies learned using MA-CPT-Q (Algo. 3), MA-CPT-Q-WS (Algo. 4), Individual Q-Learners [14] and Q-learning with weight sharing [30]. In each case, we generate 100 sample paths. We see that agents using MA-CPT-Q (blue bars) or MA-CPT-Q-WS (green bars) are able to avoid collisions, while this is not the case for the two baseline methods (orange and yellow bars).

However, fewer collisions when optimizing a CPT-metric could come at the cost of receiving a lower reward (shown as a higher accumulated cost) in Fig. 4. This insight is illustrated by the fact that there are some sample paths with high cost when agents use the MA-CPT-Q or the MA-CPT-Q-WS algorithm. A reason for this behavior could be that a risk-seeking agent may be willing to choose a low-probability action that has a higher immediate reward (low cost) over a more certain action with a lower immediate cost. As a consequence, the agent may have to trade-off between the number of

collisions (taking risk with losses) and the length of the path from start to goal (playing it safe with gains).

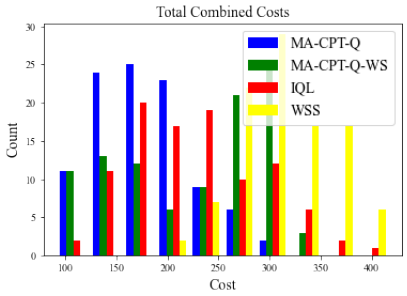

Figure 4: Comparison of total costs incurred when following policies learned using MA-CPT-Q (Algo. 3), MA-CPT-Q-WS (Algo. 4), Individual Q-Learners [14] and Q-learning with weight sharing [30]. In each case, we generate 100 sample paths. Agents following MA-CPT-Q or MA-CPT-Q-WS may sometimes incur a higher cost. A reason for this is that avoiding a collision with an obstacle or another agent might cause the agent to take a longer route to the goal state.

We also observe that agents using the MA-CPT-Q-WS algorithm may incur a higher cost than when following the MA-CPT-Q algorithm. A possible reason for this conjectured in [30] is that at the early stages of the learning procedure, agents with the fewest collisions will be deemed to have the most experience (i.e., highest rewards), and this may vary as the learning process progresses. Our observations for agents using a CPT-based algorithm are also consistent with this conjecture.

6.3 Discussion and Future Directions The results in this section demonstrate that behaviors learned by agents when using the MA-CPT-Q or MA-CPT-Q-WS algorithms is better aligned with empirical observations of behaviors of a human user [13,25]. For example, a human user placed in the same environment might sometimes prefer to take a longer route if this path will not involve confrontation with other entities in the environment (collision with other agents) or collisions with obstacles. Our experiments show that a CPT-based Q-function is sufficiently rich to model such behaviors.

We would like to remark that our objective in this paper was to develop a framework for risk-sensitive MARL in a manner that recognized preferences of humans, and to demonstrate that our approach was a reasonable solution through illustrations on relatively simple use-cases. In order to extend this framework to environments with continuous state and action spaces, we will need to explore gradient-based methods for policy learning. While these methods have been well-studied in the general MARL literature examining the optimization of an expected utility for the agents [16], their adaption to risk-sensitive MARL is still a nascent domain [17]. A promising direction of future research is to develop, analyze, and evaluate gradient-based MARL algorithms incorporating CPT-based utility and weighting functions, and compare the performance of these algorithms against benchmarks in MARL.

7 Conclusion

This paper presented a way to enable multiple reinforcement learning agents to learn behaviors that were cognizant of the preferences of human users who might also share the same environment. We used cumulative prospect theory (CPT) to model multiple human characteristics, and incorporated these into the objective function that the learning agents aimed to optimize. This paper is a natural extension to a line of recent research [26,28,29] on the integration of prospect-theoretic methods with reinforcement learning. We proposed a multi-agent CPT-based Q-learning algorithm and proved its convergence. We also examined a variant of this algorithm when agents could gather information from other agents in order to aid their learning. Our algorithms were evaluated in an environment where multiple agents had to learn behaviors to reach a target state while avoiding collisions with obstacles and with each other. Our results demonstrated that learned agent behaviors were better aligned with those shown by human users in the same environment, and significantly improved over two baseline algorithms.

References

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

[2] D. P. Bertsekas, *Dynamic Programming and Optimal Control, Vol. 1, 4th Ed.* Athena Scientific, 2017.

[3] R. Hafner and M. Riedmiller, "Reinforcement learning in feedback control," *Machine Learning*, vol. 84, pp. 137–169, 2011.

[4] V. Mnih *et al.*, "Human-level control through deep RL," *Nature*, vol. 518, no. 7540, 2015.

[5] D. Silver *et al.*, "Mastering the game of Go with DNNs and tree search," *Nature*, vol. 529, no. 7587, 2016.

[6] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[7] D. Sadigh, S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions." in *Robotics: Science and Systems*, 2016.

[8] Z. Yan and Y. Xu, "Data-driven load frequency control for stochastic power systems: A deep reinforcement

learning method with continuous action search," *IEEE Transactions on Power Systems*, vol. 34, no. 2, 2018.

[9] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse RL," *Robotics and Autonomous Systems*, vol. 114, pp. 1–18, 2019.

[10] S. A. Seshia, D. Sadigh, and S. S. Sastry, "Formal methods for semi-autonomous driving," in *ACM/EDAC/IEEE Design Automation Conference*. IEEE, 2015, pp. 1–5.

[11] N. Nikolakis, V. Maratos, and S. Makris, "A cyber physical system approach for safe human-robot collaboration in a shared workplace," *Robotics and Computer-Integrated Manufacturing*, vol. 56, pp. 233–243, 2019.

[12] B. Xiao, Q. Lu, B. Ramasubramanian, A. Clark, L. Bushnell, and R. Poovendran, "FRESH: Interactive reward shaping in high-dimensional state spaces using human feedback," in *International Conference on Autonomous Agents and MultiAgent Systems*, 2020, pp. 1512–1520.

[13] D. Kahneman and A. Tversky, "Prospect theory: An analysis of decision under risk," *Econometrica*, vol. 47, no. 2, pp. 263–292, 1979.

[14] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the International Conference on Machine Learning*, 1993, pp. 330–337.

[15] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems," *The Knowledge Engineering Review*, vol. 27, no. 1, pp. 1–31, 2012.

[16] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," *Handbook of Reinforcement Learning and Control*, pp. 321–384, 2021.

[17] W. Qiu, X. Wang, R. Yu, R. Wang, X. He, B. An, S. Obraztsova, and Z. Rabinovich, "RMIX: Learning risk-sensitive policies for cooperative reinforcement learning agents," *Advances in Neural Information Processing Systems*, vol. 34, pp. 23 049–23 062, 2021.

[18] J. Zhao, M. Yang, X. Hu, W. Zhou, and H. Li, "DQMIX: A distributional perspective on multi-agent reinforcement learning," *arXiv preprint arXiv:2202.10134*, 2022.

[19] R. T. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of banking & finance*, vol. 26, no. 7, pp. 1443–1471, 2002.

[20] M. Ahmadi, U. Rosolia, M. D. Ingham, R. M. Murray, and A. D. Ames, "Constrained risk-averse Markov decision processes," in *AAAI Conference on Artificial Intelligence*, 2021.

[21] M. P. Chapman, R. Bonalli, K. M. Smith, I. Yang, M. Pavone, and C. J. Tomlin, "Risk-sensitive safety analysis using conditional value-at-risk," *IEEE Transactions on Automatic Control*, 2021.

[22] L. Lindemann, G. J. Pappas, and D. V. Dimarogonas, "Control barrier functions for nonholonomic systems under risk signal temporal logic specifications," in *IEEE Conference on Decision and Control (CDC)*. IEEE, 2020, pp. 1422–1428.

[23] U. Schmidt, "Reference dependence in cumulative prospect theory," *Journal of Mathematical Psychology*, vol. 47, no. 2, pp. 122–131, 2003.

[24] N. C. Barberis, "Thirty years of prospect theory in economics: A review and assessment," *Journal of Economic Perspectives*, vol. 27, no. 1, pp. 173–96, 2013.

[25] A. Tversky and D. Kahneman, "Advances in prospect theory: Cumulative representation of uncertainty," *Journal of Risk and uncertainty*, vol. 5, no. 4, pp. 297–323, 1992.

[26] B. Ramasubramanian, L. Niu, A. Clark, and R. Poovendran, "Reinforcement learning beyond expectation," in *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE, 2021, pp. 1528–1535.

[27] V. S. Borkar and S. Chandak, "Prospect-theoretic Q-learning," *Systems & Control Letters*, vol. 156, no. 10, p. 105009, 2021.

[28] A. Rajabi, A. Al Maruf, B. Ramasubramanian, and R. Poovendran, "Privacy-preserving reinforcement learning beyond expectation," in *Conference on Decision and Control (CDC)*. IEEE, 2022, pp. 4706–4713.

[29] A. Al Maruf, L. Niu, B. Ramasubramanian, A. Clark, and R. Poovendran, "Cognitive bias-aware dissemination strategies for opinion dynamics with external information sources," in *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. ACM, 2023.

[30] M. N. Ahmadabadi and M. Asadpour, "Expertness based cooperative Q-learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 1, pp. 66–76, 2002.

[31] E. A. Hansen, D. S. Bernstein, and S. Zilberstein, "Dynamic programming for partially observable stochastic games," in *AAAI*, vol. 4, 2004, pp. 709–715.

[32] A. Majumdar and M. Pavone, "How should a robot assess risk? Towards an axiomatic theory of risk in robotics," in *Robotics Research*. Springer, 2020.

[33] C. Jie, L. A. Prashanth, M. Fu, S. Marcus, and C. Szepesvári, "Stochastic optimization in a cumulative prospect theory framework," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, 2018.

[34] D. Prelec, "The probability weighting function," *Econometrica*, pp. 497–527, 1998.

[35] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3, pp. 279–292, 1992.

[36] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *Journal of machine learning research*, vol. 4, no. Nov, pp. 1039–1069, 2003.

[37] A. M. Fink, "Equilibrium in a stochastic n-person game," *Journal of Science of the Hiroshima University, Series AI (Mathematics)*, vol. 28, no. 1, pp. 89–93, 1964.

[38] J. Filar and K. Vrieze, *Competitive Markov decision processes*. Springer Science & Business Media, 2012.