Toward Risk-based Optimistic Exploration for Cooperative Multi-Agent Reinforcement Learning

Jihwan Oh*
Department of Economics and Law
Korea Military Academy
Seoul, South Korea
ericoh92920@gmail.com

Joonkee Kim*
Kim Jaechul Graduate School of AI
KAIST
Seoul, South Korea
joonkeekim@kaist.ac.kr

Minchan Jeong
Kim Jaechul Graduate School of AI
KAIST
Seoul, South Korea
mcjeong@kaist.ac.kr

Se-Young Yun
Kim Jaechul Graduate School of AI
KAIST
Seoul, South Korea
yunseyoung@kaist.ac.kr

ABSTRACT

The multi-agent setting is intricate and unpredictable since the behaviors of multiple agents influence one another. To address this environmental uncertainty, distributional reinforcement learning algorithms that incorporate uncertainty via distributional output have been integrated with multi-agent reinforcement learning (MARL) methods, achieving state-of-the-art performance. However, distributional MARL algorithms still rely on the traditional ϵ-greedy, which does not take cooperative strategy into account. In this paper, we present a risk-based exploration that leads to collaboratively optimistic behavior by shifting the sampling region of distribution. Initially, we take expectations from the upper quantiles of state-action values for exploration, which are optimistic actions, and gradually shift the sampling region of quantiles to the full distribution for exploitation. By ensuring that each agent is exposed to the same level of risk, we can force them to take cooperatively optimistic actions. Our method shows remarkable performance in multi-agent settings requiring cooperative exploration based on quantile regression appropriately controlling the level of risk.

KEYWORDS

Distributional reinforcement learning; Exploration; Multi-agent learning; Uncertainty; Risk

ACM Reference Format:
Jihwan Oh*, Joonkee Kim*, Minchan Jeong, and Se-Young Yun. 2023. Toward Risk-based Optimistic Exploration for Cooperative Multi-Agent Reinforcement Learning. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Reinforcement Learning (RL) [33] has been successfully used in various domains, such as robotics, autonomous driving, video games, economy, and operations research. Multi-agent reinforcement learning (MARL) [26, 29–31], which is an extension from the single-agent setting to the multi-agent setting, is in the spotlight because it can solve the complexity of a more realistic environment than single-agent learning. However, the behaviors of MARL algorithms are often very unpredictable because the actions chosen by each agent may influence other agents. As the complexity of simulators evolves, the unpredictability makes it difficult for algorithms to approximate the exact state-action value. To address this environmental uncertainty, distributional variants of deep RL algorithms [3, 9, 10] have been adopted in MARL, leading to the state-of-the-art performance in multi-agent settings such as the StarCraft Multi-Agent Challenges (SMAC) [27]. The distributional form of the state-action value reflects the aleatoric uncertainty arising from stochastic environments, multiple agents, and variances of reward distribution. When representing state-action value as a distribution, there are two important features: *variance* and *risk*. The *variance* per action reflects the amount of uncertainty associated with parametric and intrinsic factors when an agent acts. Thus, choosing actions with high variance is considered an optimistic approach that has the potential for a high return and is used for exploration [1, 21, 23]. The concept of *risk* has its roots in economics and the stock market

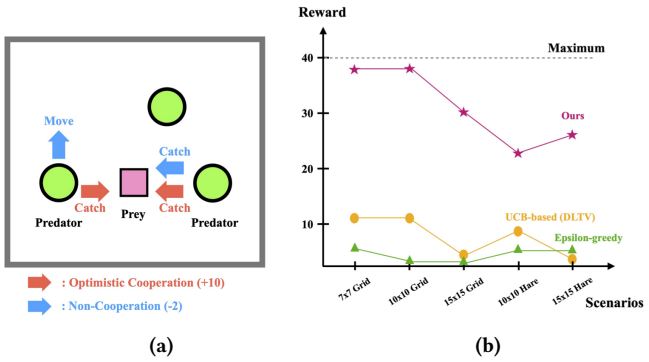


Figure 1: Motivation, Predator & Prey [6]. (a) represents how the environment works. (b) exhibits the reward per episode for each scenario. The lines are the mean of 6 random seeds.

*Equal Contribution.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

where prudent or audacious decisions are required. It has been applied to RL in which the agent selects actions based on their risks; some approaches include Risk-Sensitive RL [22] and Safe RL [12].

In this study, we employ distributional RL to address one of the most fundamental challenges in RL, the exploration & exploitation tradeoff. Exploration collects informative samples, whereas exploitation exploits the (estimated) samples or actions. In the early stages, it is more advantageous to train agents with exploratory behavior, and in the later stages start to gradually converge it towards exploitation. In multi-agent settings, the problem of exploration is more complicated due to the intrinsic uncertainty that arises from the multiple agents and unpredictable transition probability, which can be formalized using the Partially Observable MDP (POMDP) [14]. Previous works on distributional MARL algorithms either rely on ϵ-greedy [3, 9, 10, 18, 35] or UCB-based methods [8, 21, 36], both of which are inappropriate since they do not take cooperative strategies into account. However, distributional MARL algorithms still rely on the ϵ-greedy exploration [25, 29, 30], whereas numerous studies have proposed exploration strategies for distributional RL.

Figure 1 shows the environment and performance results of Predator & Prey [6] in the grid-world setting, which serves as an illustration of the importance of cooperative exploration. Predators, which are agents, get a reward when they capture prey. When two predators catch a single prey at the same time, they receive a reward of +10 and a penalty of -2 when catching a prey solely. After two predators simultaneously capture a prey, the predators are immobile and eliminated. Due to the danger of obtaining a negative reward, predators must locate and capture prey and work with other agents to maximize the reward. As shown in Figure 1(b), ϵ-greedy and UCB-based explorations (DLTV) [21] shows low performance. The result indicates that exploration methods typically employed in distributional MARL are ineffective in multi-agent environments where cooperation between agents is necessary. To overcome the unpredictability of the environment, learning to cooperate between agents requires cooperatively optimistic exploration.

In this paper, we present **R**isk-based **O**ptimistic **E**xploration (ROE), a method compatible with any existing distributional MARL algorithms, that leads to cooperatively optimistic behavior by shifting the sampling region of distribution. In this context, *distribution* is the output of any distributional RL algorithm, which is precisely the inverse CDF of the Q-value. The domain and range of the inverse CDF are referred to as *quantile fractions* and *quantile*, respectively. In the initial phase of training, for instance, we take expectations from the upper quantiles of state-action values, which lead to risky actions in pursuit of high reward, and gradually shift the sampling region of quantiles to the entire distribution. By doing so, we ensure that each agent is exposed to the same overall level of risk, compelling them to take identically optimistic actions that induce cooperation. As shown in Figure 1, our strategy, ROE, beats other considered exploration methods in which agents explore the optimal reward collaboratively. In addition, we conduct studies on the standard MARL benchmark, SMAC [27], which is a cooperative setting that is much more complicated than the Predator & Prey. Experiments are conducted using the state-of-the-art distributional MARL algorithms (DMIX, DRIMA) with our ROE as a plug-in. The results demonstrate that our strategy outperforms other exploration methods by a large margin. We summarize our contributions as follows:

- We propose a novel risk-based exploration for cooperative multi-agent settings that can be used as a plug-in for any existing distributional MARL algorithms.
- We conduct a comprehensive evaluation of our method in MARL environments and demonstrate substantial performance improvement when cooperative exploration is required.

2 BACKGROUNDS

2.1 Distributional Reinforcement Learning

In reinforcement learning, the environment is often described by the Markov Decision Process (MDP), given by a tuple $\langle X, A, P, R, \gamma \rangle$. Here, $P(x'|x,a) : X \times A \times X \to [0,1]$ is a transition probability function where x' is the next state given a current state x and action a. An agent in MDP receives rewards as the reward function $R(x,a) : X \times A \to \mathbb{R}$. $\gamma \in [0,1)$ is the reward's discount factor. The learner's goal is to find an optimal policy π maximizing the cumulative rewards $G_\pi = \sum_{t=0}^{\infty} \gamma^t R(x_t, a_t)$ with a policy $a_t \sim \pi(\cdot|x_t)$ that outputs an action distribution given a state.

Unlike traditional approaches to RL, distributional RL generates outputs as a distributional form of action. Compared to a scalar-valued reward, a distributional form of the reward gives a much richer structure to the underlying environment. Note that the scalar-valued reward is the expectation of the reward distribution. In this framework, the reward function becomes the reward distribution R, and the Q function becomes a quantile function Z. We treat the expectation $\mathbb{E}[Z(x,a)]$ as the traditional $Q(x,a)$ value. The corresponding distributional Bellman equation is defined as follows [3] :

$$\forall (x,a) \in X \times A : Z(x,a) \overset{d}{=} \mathcal{T}Z(x,a) := R(x,a) + \gamma Z(x', a'), \quad (1)$$
$$\text{where } x' \sim P(\cdot|x,a), a' \sim \pi(\cdot|x').$$

The mapping between distributions $\mathcal{T}$ is called the distributional Bellman operator [5]. This distributional RL framework with the operator $\mathcal{T}$ is being widely studied, both theoretically [3] and empirically [9].

Categorical DQN [3] gained popularity due to its superior performance in the Arcade Learning Environment (ALE) based on the Atari 2600 [4]. They output the return distribution given a state and an action by fixing the return values (known as atom or support) and approximating each return value's likelihood. The authors used the projected Kullback-Leibler (KL) divergence metric for loss functions using the shifted return values resulting from the added reward and γ. QR-DQN [10] fixes the distribution of the return as uniform and approximates return values with quantile regression. They proved that the distributional Bellman operator is a γ-contraction w.r.t. the metric $\bar{d}_p$, which is the maximal form of the Wasserstein metric W_p:

$$\bar{d}_p(Z_1, Z_2) = \sup_{x \in X, a \in A} \underbrace{\left(\int_0^1 \left| F^{-1}_{Z_1(x,a)}(\tau) - F^{-1}_{Z_2(x,a)}(\tau) \right|^p d\tau \right)^{1/p}}_{=: W_p(Z_1(x,a), Z_2(x,a))} \quad (2)$$

where the inverse CDF F_Y^{-1} of a random variable Y can be written as,

$$F_Y^{-1}(\tau) := \inf\{y \in \mathbb{R} : \tau \leq F_Y(y)\} \ . \tag{3}$$

Furthermore, they proposed the optimization method using the distributional TD-error $\delta_{\tau\tau'}$:

$$\delta_{\tau\tau'} = R(x,a) + \gamma F^{-1}_{Z_\theta(x',a')}(\tau') - F^{-1}_{Z_\theta(x,a)}(\tau)\,, \tag{4}$$

where $x' \sim P(\cdot|x,a)$ and $a' \sim \pi(\cdot|x')$. In QR-DQN, each distribution of returns per action can be defined by a linear combination of Dirac measures as follows.

$$Z_\theta(x,a) := \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i}(x,a)\,. \tag{5}$$

Here, θ_i and N represents the return value and the number of return values each. IQN [9] does not fix the probability of distribution and randomly selects the quantile fractions from a uniform distribution, $\mathcal{U}[0,1]$. Distributional RL [9, 10] uses Huber [13] quantile regression loss ρ_τ^k defined as:

$$\rho_\tau^k(\delta_{\tau\tau'}) = |\tau - \mathbb{I}\{\delta_{\tau\tau'} < 0\}| \cdot \mathcal{L}_k(\delta_{\tau\tau'}), \tag{6}$$

where

$$\mathcal{L}_k(\delta) = \begin{cases} \frac{1}{2k}\delta^2 & \text{if } |\delta| \leq k \\ |\delta| - \frac{1}{2}k & \text{otherwise} \end{cases} \tag{7}$$

Based on the distributional Bellman operator, NDQFN [36] and SPL-DQN [18] utilized a monotonic structure design to guarantee non-decreasing return values according to the arising quantile fractions. Instead of sampling quantile fractions from a distribution, FQF [35] samples quantile fractions as a parameterized model. Recently, risk-sensitive RL has been conducted based on the distributional RL due to its ability to handle quantile fractions [16].

2.2 Risk-Sensitive Policy

Generally, in risk-sensitive RL [22], risk levels can be divided into three sections: risk-averse, risk-neutral, and risk-seeking. Due to the variation in action space, a risk-sensitive policy can be read differently based on the context. Nonetheless, in this section, we will describe the general concept of risk-related policy. A risk-averse policy can be interpreted as acting with the highest state-action value among the worst-case scenarios per action. A risk-seeking policy entails selecting the same action as a risk-averse policy but based on the best-case scenario. Risk-neutral policy positions amid risk-averse and seeking policy positions.

2.3 Multi-Agent Reinforcement Learning

We now review some recent developments in deep MARL. VDN [31] considers a joint state-action value (Q_{joint}), which is just the summation of all agents' state-action values (Q_{agent}). QMIX [26], which is the most well-known algorithm in MARL, maintains monotonicity in incorporating Q_{agent} to Q_{joint}. Recently, some MARL algorithms adopted a distribution-based architecture. DMIX [30] integrated distributional RL and MARL via mean-shape decomposition, which is inspired by QMIX [26]. In DMIX, a small number of quantile fractions are sampled from $\mathcal{U}[0,1]$, resulting in a distorted uniform distribution rather than a perfect one. However, when considering the entire episode, the sampling quantile fractions approach the uniform distribution $\mathcal{U}[0,1]$. DRIMA [29] divided risk sources into cooperative and environmental risks, and injected risk levels into the agent utility function and centralized utility function differently according to the environment. They employed the architecture of the QTRAN [28] for the overall structure and the QMIX for the hypernetwork. RMIX [25] adopted the Conditional Value at Risk (CVaR) as a surrogate of joint state-action value Q_{joint} and developed a model that adaptively estimates CVaR at every step.

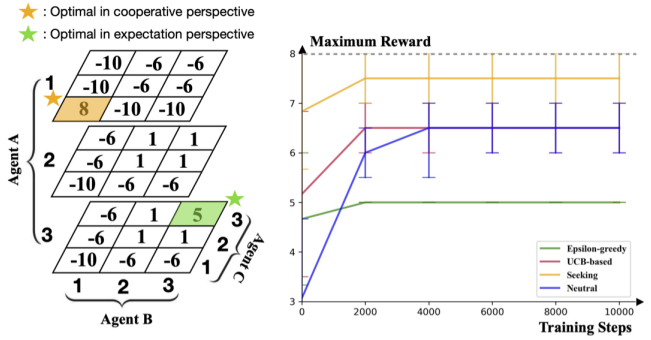
2.4 Exploration in RL

Exploration is the key problem in reinforcement learning. It has an inherent trade-off with exploitation, which is significant as it impacts the sample efficiency of RL, and can be affected by many factors such as sparse or delayed reward, large state & action space, and more. Various algorithms, such as the ϵ-greedy [33], Boltzmann exploration [32], noise perturbation [11], and intrinsic motivation [2, 7, 24], have been developed to solve this problem. After being developed using deep learning-based distributional RL [3], several distributional RL exploration methods have utilized the distributional property. DLTV [21] uses the QR-DQN algorithm [10] with optimistic action selection via the return distribution's left truncated variance. Analogous to the Upper Confidence Bound (UCB) approach in bandits literature , QR-DQN suppresses the intrinsic uncertainty by decaying the bonus such that only parametric uncertainty is utilized. The Distributional Predict Error (DPE) algorithm [36] utilizes Random Network Distillation [7] to generate two identical architectures with randomly initialized parameters and use their Wasserstein distance to measure an action's novelty given a state.

Although these methods are effective in the single-agent domain, naïvely applying them separately and independently to each agent in the MARL setting is bound to result in suboptimal performance. This is because the agents mutually influence one another, creating additional uncertainty that has to be taken into account. One way of accounting for such uncertainty is to require cooperative behavior between agents. Recently, there have been some works on ensuring cooperation between the agents. MAVEN [20] uses a hierarchical architecture to generate a shared latent vector for each agent to explore the space cooperatively. CMAE [17] creates an exploration compartment for each agent that is not shared with other agents, drastically reducing the searching space via cooperative behavior. However, none of these algorithms account for inherent uncertainty and are applied to other MARL algorithms.

3 RISK-BASED OPTIMISTIC EXPLORATION

We propose a model-agnostic risk-based optimistic exploration method for a cooperative multi-agent setting by shifting the sampling region of the state-action value's distribution. In Section 3.1, we first show the limitations of existing model-agnostic exploration methodologies and the conventional definition of risk for MARL. In Section 3.2, we discuss how our methodology works to achieve cooperative optimism by satisfying the γ-contraction in the distributional Bellman operator.



(a) (b)

Figure 2: Toy example on 1-Step Payoff game. (a) represents the matrix game, and (b) shows the reward of the each approach.

3.1 Risk in MARL

Figure 2 shows a clear performance gap between different exploration methods for our considered toy example. In this environment, the true state-action values for all agents are $Q(x, a_1) = -56$, $Q(x, a_2) = -30$, and $Q(x, a_3) = -26$ respectively. Although the maximum reward 8 is given when all agents choose a_1, for each agent, a_3 is the best action to maximize its individual reward in expectation. Therefore, unless the agents are altruistic, each agent will choose a_3, which is its individually optimistic action. In order to obtain the largest reward 8, the environment must force the agents to pick the cooperatively optimistic action, a_1. We use the DMIX algorithm in this environment and sample τ from $\mathcal{U}[0, 1]$, which is the default risk-neutral setting. As shown in Figure 2(b), ϵ-greedy exploration fails to identify the maximum reward until the end of the training and instead, obtains a suboptimal reward 5 in expectation. Moreover, it can be seen that ϵ-greedy exploration even performs worse than the risk-neutral setting, which suggests that the ϵ-greedy approach hinders cooperative exploration. The multi-agent version of DLTV (UCB-based method) receives a reward of 8 half of the time and a reward of 5 the other half of the time. This is because DLTV compels agents to choose optimistic, non-cooperative actions, which results in optimal value when they are fortunate. We also consider a simple risk-based method (risk-seeking) for optimistic action.

After the quantile fractions' sampling region is specified, the risk-based distributional RL selects an action as follows.

$$a^* = \underset{a \in A}{\operatorname{argmax}} \underset{\tau \sim \mathcal{U}[\alpha,\beta]}{\mathbb{E}} \left[F^{-1}_{Z(x,a)}(\tau) \right] \tag{8}$$

Usually, α and β are set to 0 and 1 each to utilize full distribution, but this isn't always the case. For instance, assuming that we want $0.5 < \alpha < \beta$ and that the distribution is symmetric about 0.5, we get a general inequality as follows:

$$\mathbb{E}\left[Z(x, a)\right] \leq \underset{\tau \sim \mathcal{U}[\alpha,\beta]}{\mathbb{E}} \left[F^{-1}_{Z(x,a)}(\tau) \right] \tag{9}$$

If the given F^{-1}, state and action are identical, the agent overestimates the state-action value as shown in Equation (9) with upper quantile fractions. Therefore, the agents now choose risk-seeking (optimism) action, leading to superior performance than risk-neutral policy when cooperative behavior is desired as shown in Figure 2. Here, we set α and β to 0.75 and 1.0 to implement a risk-seeking policy. Indeed, as shown in Figure 2(b), the risk-seeking approach significantly outperforms ϵ-greedy and DLTV, reaching the maximum reward more often and having a greater reward in expectation as well.

Although the previous toy example suggests that risk-seeking always yields superior performance via the effect of cooperative optimism, this isn't generally the case in a more complex and long-episodic environment. In such environments, in contrast to the 1-step payoff game, continually seeking a high reward is not exploitation. As the long-term episode requires the agents to decide their actions consecutively, the only-seeking method's cooperative strategy is broken. The agents have to exploit the estimated samples from the optimistic actions rather than explore only seeking behavior.

3.2 Cooperative Optimism with Risk Scheduling

We propose ROE, **R**isk-based **O**ptimistic **E**xploration, which addresses the difficulties of multi-agent environments requiring cooperation as discussed in Section 3.1. We achieve cooperative optimism in a multi-agent setting by endowing each agent with an identical risk level, hence inducing similar behaviors across the agents. By imposing a high-risk level at the initial phase, (e.g., $\tau \sim \mathcal{U}[0.75, 1]$), we make the agents choose informative action in a cooperative manner. We then gradually update the sample region, starting from the upper domain $\tau \sim \mathcal{U}[0.75, 1]$ to the full domain $\tau \sim \mathcal{U}[0, 1]$. Lastly, the agents exploit the estimated samples of the entire distribution.

We allow agents to explore cooperatively optimistic actions, gradually exploiting the optimistically estimated samples using Equation (8) where α and β adjust the risk levels (confidence bound of distribution) and keep changing through the scheduling steps as illustrated in Algorithm 1.

3.2.1 Dynamics of risk-scheduling. ROE shifts risk levels from the *seeking* to specific levels. Like to previous works [3, 15], our method is equivalent to iterating a finite sequence of operators $\{\mathcal{T} \circ \Pi_{\alpha_t,\beta_t}\}_{t=1}^{T}$, where Π_{α_t,β_t} is the uniform projection on the quantile range $[\alpha_t, \beta_t]$. We discuss the contraction property of the distributional Q function and apply the distributional optimality operator $\mathcal{T} = \mathcal{T}^{\pi}$ for a greedy policy π [3].

We first note the non-expansive property of the projection operator in the following Lemma 1.

Lemma 1 (Non-expansiveness). Let $\Pi_{\alpha,\beta}$ $(0 \leq \alpha < \beta \leq 1)$ be the transformation on the random variable, defined by the quantile function or inverse CDF as

$$F^{-1}_{\underline{\Pi_{\alpha,\beta} Z(x,a)}}(\tau) = F^{-1}_{Z(x,a)}\left((\beta - \alpha)\tau + \alpha\right), \quad \text{where } \tau \in [0, 1],\ (x, a) \in X \times A.$$

Then the $\Pi_{\alpha,\beta}$ is non-expansive on the metric $\bar{d}_\infty$:

$$\bar{d}_\infty(Z_1, Z_2) = \sup_{x \in X, a \in A} \operatorname*{esssup}_{\tau \in [0,1]} \left| F^{-1}_{Z_1(x,a)}(\tau) - F^{-1}_{Z_2(x,a)}(\tau) \right|.$$

Algorithm 1 ROE [Linear scheduling]

Require:

$k \leftarrow$ scheduling time steps

$\omega_0 \leftarrow$ initial risk level, $\omega_k \leftarrow$ final risk level

We set risk level interval to [-1, 1] for the convenience of computation. Risk level 1 (extreme seeking), 0.5, 0 (neutral), -0.5, -1 (extreme averse) means sampling quantile fractions from $\mathcal{U}[1,1], \mathcal{U}[0.5,1], \mathcal{U}[0,1], \mathcal{U}[0,0.5], \mathcal{U}[0,0]$ each. Therefore, if we set $\omega_0 = 1$ and $\omega_k = 0$, then it means that I will schedule the risk level from risk-seeking to neutral.

Ensure:

$\omega_t \leftarrow$ current risk level $(= [\alpha_t, \beta_t])$, $0 \le \alpha_t \le \beta_t \le 1$

Risk-scheduling size δ is $\delta = \frac{\omega_0 - \omega_k}{k}$

Store random trainsition (x_t, a_t, r_t, x_{t+1}) in ReplayBuffer $\mathcal{D}$ for short learning time.

$\omega_t \leftarrow \omega_0$

while $t < T$ **do**

  Select an action, $a_t = \text{argmax}_{a \in A} \mathbb{E}_{\tau \sim \mathcal{U}[\alpha_t, \beta_t]}[F^{-1}_{Z(x,a)}(\tau)]$

  Execute an action a_t and observe r_t and x_{t+1}

  Store transition (x_t, a_t, r_t, x_{t+1}) in ReplayBuffer $\mathcal{D}$

  Sample transition batch from ReplayBuffer $\mathcal{D}$

  $\mathcal{L}_k = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{N} \rho^k_{\tau_i}(\delta'^t_{\tau_i \tau_j}), \quad \tau_i, \tau'_j \sim \mathcal{U}[\alpha_t, \beta_t]$

  if $t \le k$ **then**

    $\omega_{t+1} \leftarrow \omega_t - \delta$

  else

    $\omega_{t+1} \leftarrow \omega_k$

  end if

end while

Therefore, if the distributional Bellman operator with greedy policy $\mathcal{T}$ is a γ-contraction, so is $\mathcal{T} \circ \Pi_{\alpha,\beta}$ on $\bar{d}_\infty$, for fixed α and β. Furthermore, by the Banach fixed point theorem, there also exists a unique fixed point $Z_{\alpha,\beta}$ for $\mathcal{T} \circ \Pi_{\alpha,\beta}$. Each fixed point, which is precisely the distributional Q function, reflects the various risk level by allowing agents to behave differently.

From these observations, we propose a scheduling method to allow the agents to various risk-sensitivity. Since the iterating operator changes with time, the procedure is governed by the temporal evolution of the operator. This is especially true when α_t and β_t change at a rate of $o(T^{-1})$. In such cases, a mere convergence result does not provide much information. Instead, we show that the distance between the t-th step and the fixed point $Z^*_{\alpha_t,\beta_t}$ can be bounded as shown in the following proposition :

Proposition 1. Let us consider the iterative process $Z_t \leftarrow \mathcal{T} \circ \Pi_{\alpha_t,\beta_t}(Z_{t-1})$, and denote $Z^*_{\alpha_t,\beta_t}$ as the unique fixed point of $\mathcal{T} \circ \Pi_{\alpha_t,\beta_t}$. Then we have the upper bound between the distance of the t-th state and the fixed point of t-th operator as:

$$\bar{d}_\infty(Z_t, Z^*_{\alpha_t,\beta_t}) \le \sum_{i=1}^{t-1} \gamma^{t-i} \bar{d}_\infty(Z^*_{\alpha_i,\beta_i}, Z^*_{\alpha_{i+1},\beta_{i+1}}) + \gamma^t \bar{d}_\infty(Z_0, Z^*_{\alpha_1,\beta_1}).$$

The upper bound is a weighted combination of the $\bar{d}_\infty$-distance between the neighboring fixed points. Intuitively, recent information has a more significant influence, which exponentially decreases with its age by a factor of γ, the discount factor. One important observation is that if (α_t, β_t) changes moderately towards (α, β), Z_t remains close to $Z^*_{\alpha,\beta}$ because the distance between the fixed points will be close.

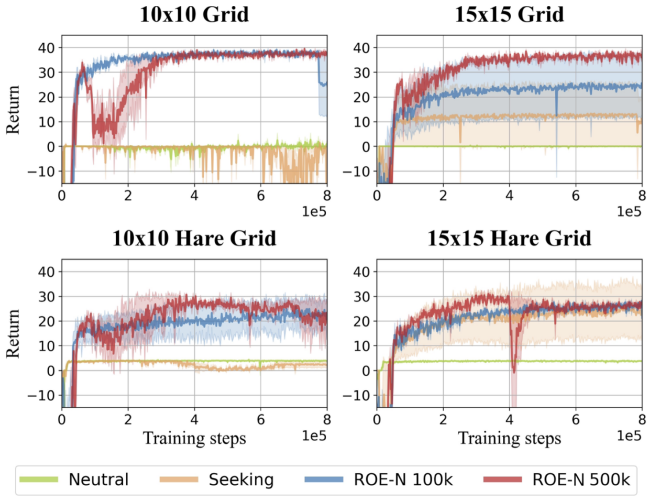


Figure 3: Episode return of DMIX in Predator & Prey experiments. The lines are the mean of 3 random seeds with shaded areas representing a confidence interval of 25% to 75%. The numbers represent risk-scheduling steps.

4 EXPERIMENTS

For the experiments, we consider two variants of ROE: ROE-N refers to ROE that adjusts the risk level from risk-seeking to risk-neutral, and ROE-A refers to ROE that adjusts from risk-seeking to risk-averse. We evaluate ROE-N and ROE-A in two cooperative multi-agent settings with high aleatoric uncertainties. One is a Predator & Prey environment and the other is the Starcraft Multi-Agent Challenges (SMAC). As an ablation study, we also consider a single-agent setting that does not require a cooperative strategy.

4.1 Environments

Predator & Prey As dealt with previously in the Introduction section, Predator & Prey [6] is a grid environment in which the 8 predators (agents) must capture 8 prey cooperatively. The environment has inherent stochasticity as follows: with probability 0.1 an "up" action will not be executed, and the transition of each predator is governed by a transition probability kernel $P(x'|x, a)$. Each prey begins each episode at an arbitrary point and behaves in a random manner, resulting in an inability to remember sequences for agents. Moreover, agents can observe only within two grids from them, which makes this environment a POMDP. The environment thus requires cooperative strategies and optimistic exploration to capture the prey. Additionally, in the harder scenario `Hare Grid`, there exist rabbits that are similar to prey in the way of giving rewards but provide reward 1. Such rabbits are used as deceptive reward signals which hinder predators from capturing prey.

StarCraft Multi-Agent Challenges (SMAC) For more complex POMDP multi-agent settings, we conduct experiments on SMAC environments [27], the standard cooperative multi-agent RL

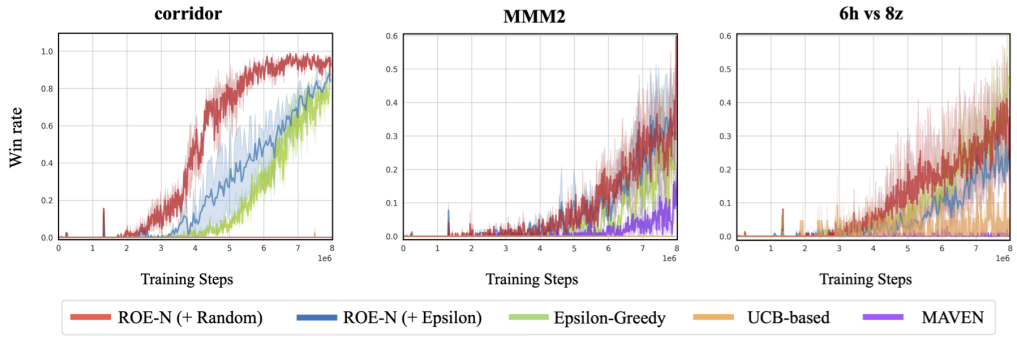


Figure 4: Comparison of exploration algorithms in Superhard scenarios. The lines are the mean of 3 random seeds using five parallel training with shaded areas representing a confidence interval of 25% to 75%. Baseline architecture is DRIMA.

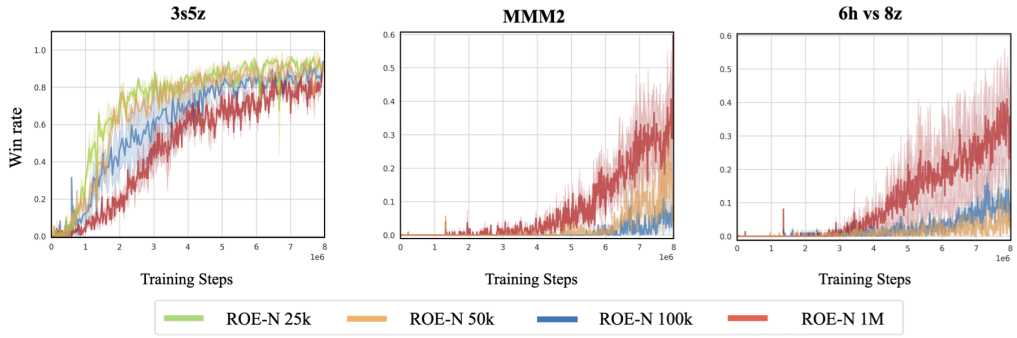


Figure 5: Performance sensitiveness of our method (ROE) according to the risk-scheduling steps on *Easy* scenario (3s5z) and *Superhard* scenarios (MMM2 & 6h vs 8z). The lines are the mean of 3 random seeds using five parallel training with shaded areas representing a confidence interval of 25% to 75%. Baseline architecture is DRIMA.

benchmark, with a focus on micro-management challenges. Each SMAC environment consists of allies and enemies, each evaluating their win rate. Allies are controlled by the MARL algorithms, while enemies are controlled by the original StarCraftII agents with a difficulty level 7 out of 10. Allies receive the episode's reward of 200 when they win a battle, as well as small rewards of 10 for killing an enemy and a payout equal to the amount of damage they dealt to adversaries. To win a battle, agents must cooperate among themselves to manage their group behavior, like *focusing fire* while not overkilling the enemies, or *kiting* to lure the enemies and kill them one by one. We report the results for *SuperHard* scenario, where the importance of cooperation is crucial in winning.

Atari We evaluated the validity of our method in an Atari game [4, 19], a single-agent setting where intrinsic uncertainty is very low and cooperation is not necessary. Here, the environment is fully deterministic, and the reward consists of {-1, 0, 1}. Specifically, the experiments are conducted in situations where complex exploration is needed [34].

4.2 Implementation

For risk-based optimistic exploration, we shift the sampling region of distribution with *linear* scheduling. We plugged in our ROE method to IQN, DMIX, and DRIMA. For IQN and DMIX, we define risk-averse, risk-neutral, and risk-seeking to be the sampling quantile fractions from $\mathcal{U}[0, 0.25]$, $\mathcal{U}[0, 1]$, and $\mathcal{U}[0.75, 1]$, respectively; for DRIMA, they were set to be the sampling quantile fractions from $\mathcal{U}[0, 0.1]$, $\mathcal{U}[0.4, 0.5]$, and $\mathcal{U}[0.9, 1.0]$. To schedule risk in IQN and DMIX from seeking to neutral, we initialized $(\alpha, \beta) = (0.99, 1.0)$ in Equation 8 at first, for more optimistic exploration, which is generally set to (0.75, 1) for risk-seeking policy. Then, quantile fractions 0.99 (α) is linearly decayed to 0. When α becomes 0, the risk level is positioned at risk-neutral, which samples quantile fractions from $\mathcal{U}[0, 1]$. In DRIMA, we linearly shift the quantile sampling index from $\mathcal{U}[0.9, 1.0]$ to $\mathcal{U}[0.4, 0.5]$ by an increment of 0.1 to correspond to the architecture of the underlying algorithm. In SMAC environment, we evaluate both ROE-N and ROE-A. This is because we have observed that, depending on the risk level of the agents, SMAC displays distinct behaviors that directly influence the win rate.

4.3 Results

In MARL experiments with high uncertainty levels where cooperation is necessary, our risk-based exploration yields a significant performance advantage over ϵ-greedy, UCB-based exploration, and static risk level-based approaches.

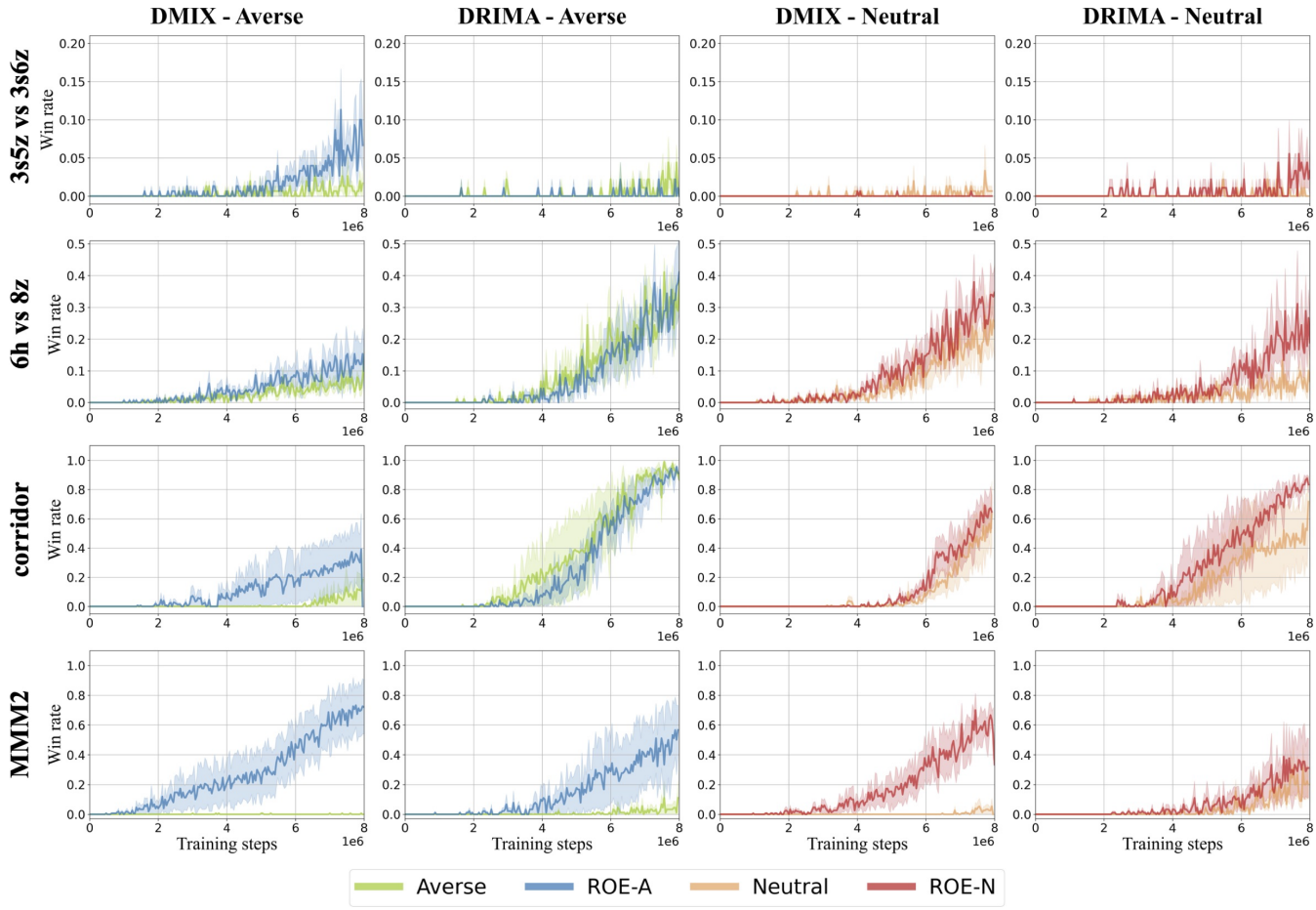


Figure 6: Win-rate Results of DMIX and DRIMA in Superhard scenarios. The label of X-axis and Y-axis represent algorithm - risk level and scenarios respectively. The lines are the mean of 5 random seeds in DMIX, 3 random seeds in DRIMA using five parallel training with shaded areas representing a confidence interval of 25% to 75%.

Predator & Prey We compare our method with static risk level-based approaches. Figure 3 shows the training curves of each algorithm. Predators plugged with our method effectively resolve the problem that static predators could not. In environments `10x10 Grid` and `15x15 Grid`, which lack deceptive reward compared to `Hare Grid`, risk-neutral predators could evade the negative reward that results from solely capturing, but they do not learn how to get a greater reward. Risk-seeking predators demonstrate moderately superior or even worse than risk-neutral predators. However, our method initially receives negative rewards but has cooperative optimism, which will be decayed to find better rewards, so they learn the appropriate methods and employ them effectively. In the setting that has a deceptive reward, `Hare`, shows similar results. Risk-neutral predators only capture rabbits (deceptive rewards), hence never capturing prey. It is easy for static risk-neutral predators to learn how to take rabbits but difficult to learn how to capture prey. Although risk-seeking predators perform similarly to ROE predators in the `15x15 Hare Grid`, they do poorly in the `10x10 Hare Grid`. However, predators appear to perform well with our ROE that maximizes rewards in all `Hare Grid` while effectively avoiding rabbits. This phenomenon illustrates that our methodologies are robustly operational, even in smaller or more challenging maps (i.e., `10x10 Hare Grid`) where it is easier to receive deceptive rewards.

In addition, for a more detailed explanation for comparing exploration methods, we use the experiment results in Introduction section. As shown in Figure 1(b), our method outperforms the other exploration methods with a significant performance gap. The failure of ϵ-greedy exploration in this environment is due to the random exploration's discontinuity of preferable actions and using only expectation value to choose an action. UCB-based exploration demonstrates better exploration (sometimes reach to maximum reward when fortunate) than the ϵ-greedy method, but it exhibits most of the failure in getting rewards. This is because UCB-based exploration, choosing based on the variance of reward distribution, yields optimistic but non-cooperative action.

StarCraft Multi-Agent Challenges Our method results in considerable performance gains in SMAC. We compare the exploration methods (ϵ-greedy, UCB-based, MAVEN[20]), which are applicable to any algorithms (except MAVEN) in *Super Hard* scenarios where

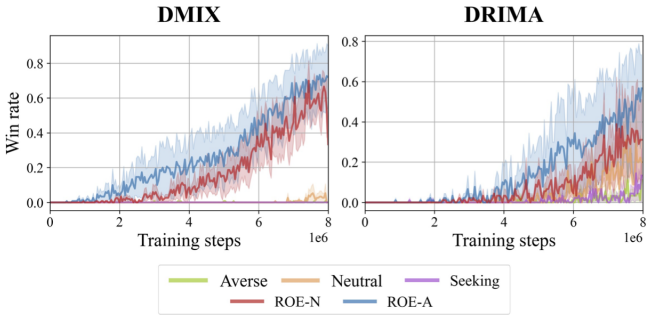


Figure 7: Win-rate Results of MMM2 scenario in SMAC. The lines are the mean of 3 (left) and 5 (right) random seeds with shaded areas representing a confidence interval of 25% to 75%.

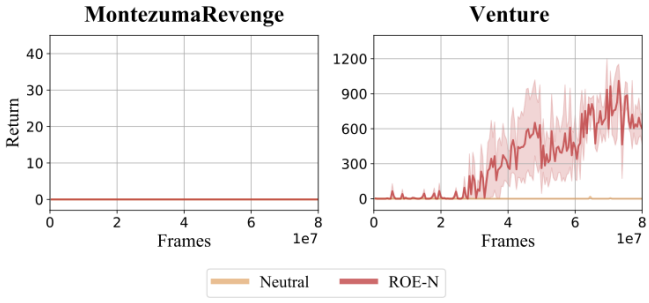


Figure 8: Episode return of IQN in Atari. The lines are the mean of 3 different risk-scheduling time steps with shaded areas representing a confidence interval of 25% to 75%.

hard exploration is required. Additionally, to make sure monotonicity in the inverse CDF function (quantile function), we collect random samples in the very initial training phase for 50k steps by random or ϵ-greedy action selector. For long-horizontal exploration, we searched exploration step in {50k, 100k, 1M} for our method and ϵ-greedy exploration and showed the best performance among them. The hyperparameters used in UCB-based and MAVEN is that showing the best performance in their papers. As shown in Figure 4, ROE with random (purple line) shows the best performance. Also, we compare ROE with a static risk-neutral, averse policies with comprehensive experiments. As depicted in Figure 6, our learning curve converges faster and obtains a higher win rate in the majority of scenarios compared to static risk policies.

The reason of the performance in Figure 4,6 is that ROE collects the merits of risk-seeking and other risk-based policies by scheduling the risk levels in SMAC. Risk-seeking policies at an early stage enable allies to explore and identify cooperative winning strategies, such as running away for a moment or moving to weakened enemies to focus fire, more quickly by encouraging them to take cooperatively optimistic actions. In contrast, risk-averse policies generally encourage allies to focus mainly on attack, which is the best course of action in the worst-case scenario. Since decaying the risk level controls this trade-off effectively, ROE could achieve the best performance among our experiments.

As demonstrated in Figure 4, similar to the results in Predator & Prey, the MMM2 environment requires exploration, but learning is challenging with naive optimistic actions from risk-seeking policy. However, the ROE is able to effectively balance exploration-exploitation by adjusting the risk level over time in the more complex *MMM2* environment, allowing for the identification of optimal strategies.

Sensitiveness of Scheduling There should be proper exploration and exploitation steps to solve the complexity in RL environments. Here, we discuss the scheduling time of altering the region of quantile fractions, which has an impact on the exploration & exploitation trade-off. Risk-seeking has the effect of exploration to search state or action, which results in a greater reward, but shifting the sampling region of distribution to the entire distribution shows exploitation based on the prior knowledge of distribution. As depicted in Figure 4 (mid, right), the longer the scheduling period is, the higher the performance in *Super hard* scenarios. In contrast, the longer the scheduling time in the *Easy* scenario (Figure 4 left), which involves more exploitation than exploration, the worse the performance. Therefore, our dynamics of risk play a role in exploration and exploitation trade-offs, and appropriate risk scheduling steps are required.

Without Aleatoric Uncertainty Although our method mainly focuses on addressing cooperative multi-agent environment, Figure 8 shows our method's performance in a deterministic single-agent environment, Atari, which requires no cooperation. We conduct experiments on hard exploration games, Montezuma's Revenge, and Venture. In the early stages of training in sparse reward environments, such as Venture, our risk-based optimistic exploration displays excellent exploration. The reason for the improvements is a bit different but similar to other exploration approaches in distributional RL [21, 36] with respect to considering the upper confidence of distribution. However, we need an intrinsic reward from distributional output to solve notably spare reward setting, such as Montezuma's Revenge.

5 CONCLUSION & FUTURE WORK

Endowing identical risk-seeking levels to agents makes them behave in a cooperatively optimistic manner. Then shifting the sampling region of distribution to the entire distribution or lower region of distribution enables the agents to utilize the exploratory samples. Experiments have demonstrated that ROE is effective in that it enhances the model's learning speed and improves final performance significantly more than other exploration techniques under aleatoric uncertainty for cooperative settings. One important future work is to develop risk-based exploration for competitive environments.

ACKNOWLEDGMENTS

This work was conducted by Center for Applied Research in Artificial Intelligence(CARAI) grant funded by Defense Acquisition Program Administration(DAPA) and Agency for Defense Development(ADD) (UD190031RD).

REFERENCES

[1] Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research* 3, Nov (2002), 397–422.

[2] Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. 2016. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems* 29 (2016).

[3] Marc G Bellemare, Will Dabney, and Rémi Munos. 2017. A distributional perspective on reinforcement learning. In *International Conference on Machine Learning*. PMLR, 449–458.

[4] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research* 47 (jun 2013), 253–279.

[5] Richard Bellman. 1966. Dynamic programming. *Science* 153, 3731 (1966), 34–37.

[6] Wendelin Böhmer, Vitaly Kurin, and Shimon Whiteson. 2020. Deep coordination graphs. In *International Conference on Machine Learning*. PMLR, 980–991.

[7] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2018. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894* (2018).

[8] Tae Hyun Cho, Sungyeob Han, Heesoo Lee, Kyungjae Lee, and Jungwoo Lee. 2022. Distributional Perturbation for Efficient Exploration in Distributional Reinforcement Learning. https://openreview.net/forum?id=rGg-Qcyplgq

[9] Will Dabney, Georg Ostrovski, David Silver, and Rémi Munos. 2018. Implicit quantile networks for distributional reinforcement learning. In *International conference on machine learning*. PMLR, 1096–1105.

[10] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. 2018. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[11] Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. 2017. Noisy networks for exploration. *arXiv preprint arXiv:1706.10295* (2017).

[12] Javier Garcıa and Fernando Fernández. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16, 1 (2015), 1437–1480.

[13] Peter J Huber. 1992. Robust estimation of a location parameter. In *Breakthroughs in statistics*. Springer, 492–518.

[14] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101, 1-2 (1998), 99–134.

[15] Ramtin Keramati, Christoph Dann, Alex Tamkin, and Emma Brunskill. 2020. Being Optimistic to Be Conservative: Quickly Learning a CVaR Policy. In *AAAI*.

[16] Shiau Hong Lim and Ilyas Malik. 2022. Distributional Reinforcement Learning for Risk-Sensitive Policies. In *Advances in Neural Information Processing Systems*.

[17] Iou-Jen Liu, Unnat Jain, Raymond A Yeh, and Alexander Schwing. 2021. Cooperative exploration for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*. PMLR, 6826–6836.

[18] Yudong Luo, Guiliang Liu, Haonan Duan, Oliver Schulte, and Pascal Poupart. 2021. Distributional Reinforcement Learning with Monotonic Splines. In *International Conference on Learning Representations*.

[19] Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. 2018. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research* 61 (2018), 523–562.

[20] Anuj Mahajan, Tabish Rashid, Mikayel Samvelyan, and Shimon Whiteson. 2019. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems* 32 (2019).

[21] Borislav Mavrin, Hengshuai Yao, Linglong Kong, Kaiwen Wu, and Yaoliang Yu. 2019. Distributional reinforcement learning for efficient exploration. In *International conference on machine learning*. PMLR, 4424–4434.

[22] Ralph Neuneier and Oliver Mihatsch. 1998. Risk sensitive reinforcement learning. *Advances in Neural Information Processing Systems* 11 (1998).

[23] Nikolay Nikolov, Johannes Kirschner, Felix Berkenkamp, and Andreas Krause. 2018. Information-directed exploration for deep reinforcement learning. *arXiv preprint arXiv:1812.07544* (2018).

[24] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*. PMLR, 2778–2787.

[25] Wei Qiu, Xinrun Wang, Runsheng Yu, Rundong Wang, Xu He, Bo An, Svetlana Obraztsova, and Zinovi Rabinovich. 2021. RMIX: Learning Risk-Sensitive Policies forCooperative Reinforcement Learning Agents. *Advances in Neural Information Processing Systems* 34 (2021).

[26] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 4295–4304.

[27] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043* (2019).

[28] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*. PMLR, 5887–5896.

[29] Kyunghwan Son, Junsu Kim, Yung Yi, and Jinwoo Shin. 2021. Disentangling Sources of Risk for Distributional Multi-Agent Reinforcement Learning. (2021).

[30] Wei-Fang Sun, Cheng-Kuang Lee, and Chun-Yi Lee. 2021. DFAC framework: Factorizing the value function via quantile mixture for multi-agent distributional q-learning. In *International Conference on Machine Learning*. PMLR, 9945–9954.

[31] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. 2017. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296* (2017).

[32] Richard S. Sutton. 1990. Integrated Architectures for Learning, Planning, and Reacting Based on Approximating Dynamic Programming. In *In Proceedings of the Seventh International Conference on Machine Learning*. Morgan Kaufmann, 216–224.

[33] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

[34] Adrien Ali Taiga, William Fedus, Marlos C Machado, Aaron Courville, and Marc G Bellemare. 2021. On bonus-based exploration methods in the arcade learning environment. *arXiv preprint arXiv:2109.11052* (2021).

[35] Derek Yang, Li Zhao, Zichuan Lin, Tao Qin, Jiang Bian, and Tie-Yan Liu. 2019. Fully parameterized quantile function for distributional reinforcement learning. *Advances in neural information processing systems* 32 (2019).

[36] Fan Zhou, Zhoufan Zhu, Qi Kuang, and Liwen Zhang. 2021. Non-decreasing Quantile Function Network with Efficient Exploration for Distributional Reinforcement Learning. *arXiv preprint arXiv:2105.06696* (2021).