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:

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.

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.

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.

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 $\epsilon$-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 $\epsilon$-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 capturing a prey solely. After two predators simultaneously capture a prey, the predators are immune 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), $\epsilon$-greedy and UCB-based explorations (DLTV) [21] show 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 Risk-based Optimistic Exploration (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 (DMMX, 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 $(X, A, P, R, \gamma)$. Here, $P(x'|x, a) : X \times A \times X \rightarrow [0, 1]$ 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 $R(x, a) : X \times A \rightarrow \mathbb{R}$. $\gamma \in [0, 1)$ is the reward’s discount factor. The learner’s goal is to find an optimal policy $\pi$ 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]:

$$\forall (x, a) \in X \times A : Z(x, a) \overset{d}{=} T Z(x, a) := R(x, a) + \gamma Z(x', a'),$$

where $x' \sim P(\cdot|x, a), a' \sim \pi(\cdot|x').$

The mapping between distributions $T$ is called the distributional Bellman operator $[5]$. This distributional RL framework with the operator $T$ 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 $\gamma$. 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 $\gamma$-contraction w.r.t. the metric $d_p$, which is the maximal form of the Wasserstein metric $W_p$:

$$d_p(Z_1, Z_2) = \sup_{x \in X, a \in A} \left( \int \left( F^{-1}_{Z_1(x, a)}(\tau) - F^{-1}_{Z_2(x, a)}(\tau) \right)^p \, d\tau \right)^{1/p}$$

$$= W_p(Z_1(x, a) | Z_2(x, a))$$

(2)
where the inverse CDF \( F_Y^{-1} \) of a random variable \( Y \) can be written as,
\[
F_Y^{-1}(t) := \inf \{ y \in \mathbb{R} : t \leq F_Y(y) \}.
\]
Furthermore, they proposed the optimization method using the distributional TD-error \( \delta_{tr}^{\gamma} \):
\[
\delta_{tr}^{\gamma}(x, a, s', a', r, \gamma) = F_{Z_\gamma}(s', a', r) Y - F_{Z_\gamma}(s, a, r),
\]
where \( Z_\gamma \) is defined later. IQN [9] does not fix the probability of distribution and considers the entire episode, the sampling quantile fractions approach the uniform distribution. RMIX [25] adopted a distribution-based architecture. Spl-DQN [18] utilized a monotonic structure design to guarantee risk-averse and seeking policy positions. Risk-neutral policy positions amid 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 entails selecting the same action as a risk-averse policy but differently based on the context. Risk-neutral policy positions amid risk-averse and seeking policy positions.

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 \( \epsilon \)-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 \( \gamma \)-contraction in the distributional Bellman operator.
which is the default risk-neutral setting. As shown in Figure 2(b), the maximum reward is 8 given when all agents choose a1, for each agent, a3 is the best action to maximize its individual reward in expectation. Therefore, unless the agents are altruistic, each agent will choose a3, 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, a1. We use the DMIX algorithm in this environment and sample a from U[0, 1], which is the default risk-neutral setting. As shown in Figure 2(b), e-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 e-greedy exploration performs worse than the risk-neutral setting, which suggests that the e-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:

\[ a^* = \arg \max_{a \in A} \mathbb{E}_{\tau \sim \mathcal{U}[\alpha, \beta]} \left[ F^{-1}_{Z(x,a)}(\tau) \right] \]  

(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 < α < β and that the distribution is symmetric around 0.5, we get a general inequality as follows:

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

(9)

If the given \( F^{-1} \), state and action are identical, the agent over-estimates 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 \( \alpha = 0.75 \) and \( \beta = 1.0 \) to implement a risk-seeking policy. Indeed, as shown in Figure 2(b), the risk-seeking approach significantly outperforms e-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, Risk-based Optimistic Exploration, 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, \( \tau \sim \mathcal{U}[0.75, 1] \), 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 \( \alpha \) and \( \beta \) 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 \( \text{seeking} \) to specific levels. Like to previous works [3, 15], our method is equivalent to iterating a finite sequence of operators \( \{T \circ \Pi_{\alpha_i, \beta_i}\}_{i=1}^T \), where \( \Pi_{\alpha_i, \beta_i} \) is the uniform projection on the quantile range \( [\alpha_i, \beta_i] \). We discuss the contraction property of the distributional Q function and apply the distributional optimality operator \( T = T^\pi \) for a greedy policy \( \pi \) [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

\[ F^{-1}_{\Pi_{\alpha, \beta} Z(x,a)}(\tau) = F^{-1}_{Z(x,a)}((\beta - \alpha)\tau + \alpha), \]

where \( \tau \in [0, 1], (x,a) \in X \times A \).

Then the \( \Pi_{\alpha, \beta} \) is non-expansive on the metric \( \hat{d}_{oo} \):

\[ \hat{d}_{oo}(Z_1, Z_2) = \sup_{x \in X, a \in A} \esssup_{\tau \in [0, 1]} \left| F^{-1}_{Z_1(x,a)}(\tau) - F^{-1}_{Z_2(x,a)}(\tau) \right|. \]
Algorithm 1 ROE [Linear scheduling]
\begin{algorithm}
\textbf{Require:}
\begin{enumerate}
\item $k \leftarrow$ scheduling time steps
\item $\omega_0 \leftarrow$ initial risk level, $\omega_k \leftarrow$ final risk level
\end{enumerate}
\textbf{Ensure:}
\begin{enumerate}
\item $\omega_t \leftarrow$ current risk level \((\equiv [a_t, \beta_t])\), \(0 \leq a_t \leq \beta_t \leq 1\)
\item Risk-scheduling size $\delta$ is $\delta = \frac{\omega_0 - \omega_k}{k}$
\item Store random transition \((x_t, a_t, r_t, x_{t+1})\) in ReplayBuffer $D$ for short learning time.
\item $\omega_t \leftarrow \omega_0$
\item \textbf{while} $t < T$ \textbf{do}
\begin{enumerate}
\item Select an action, $a_t = \arg\max_{a \in A} \mathbb{E}_{z \sim \mathcal{U}(a_t, \beta_t)} \left[ F_{z(x,a)}(t) \right]$\end{enumerate}
\textbf{end while}
\end{algorithm}

Therefore, if the distributional Bellman operator with greedy policy $T$ is a $\gamma$-contraction, so is $T \circ \Pi_{t,a,\beta}$ on $\tilde{d}_\omega$, for fixed $\alpha$ and $\beta$. Furthermore, by the Banach fixed point theorem, there also exists a unique fixed point $Z^*_{t,a,\beta}$ for $T \circ \Pi_{t,a,\beta}$. 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 $a_t$ and $\beta_t$ 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 $Z^*_{t,a,\beta}$ can be bounded as shown in the following proposition:

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

$$\tilde{d}_\omega(Z_t, Z^*_{t,a,\beta}) \leq \sum_{i=1}^{t-1} \gamma^{t-i} \tilde{d}_\omega(Z^*_{i-1, a_i, \beta_i}, Z^*_{i, a_i, \beta_i}) + \gamma^t \tilde{d}_\omega(Z_0, Z^*_{0, a_0, \beta_0}).$$

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

### 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.
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 ($\alpha$) is linearly decayed to 0. When $\alpha$ 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 $\epsilon$-greedy, UCB-based exploration, and static risk level-based approaches.
**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 10×10 Grid and 15×15 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 15×15 Hare Grid, they do poorly in the 10×10 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., 10×10 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 $\epsilon$-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 $\epsilon$-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 ($\epsilon$-greedy, UCB-based, MAVEN[20]), which are applicable to any algorithms (except MAVEN) in Super Hard scenarios where

![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%.

---

Session 5A: Multiagent Reinforcement Learning III  
AAMAS 2023, May 29–June 2, 2023, London, United Kingdom
we compare ROE with a static risk-neutral, averse policies with exploration step in \{50k, 100k, 1M\} for our method. Figure 4, ROE with random (purple line) shows the best performance. Also, Prey, the MMM2 environment requires exploration, but 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 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 settings, 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


