Q-FAC: Quantile Value Function Factorization for Multi-Robot Moving Target Search with High Percentile Confidence

Anonymous Submission

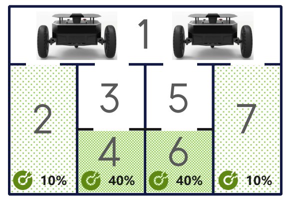
Abstract

Existing multi-robot search (MuRS) formulations typically target minimizing the expected search time, which may be inadequate in the presence of long-tailed uncertainties. In this paper, we formulate a new MuRS problem, termed multi-robot quantile search (MuRQS), where a team of robots cooperatively search for a moving target by minimizing the high-percentile search time. However, optimizing such a quantile-based search objective is challenging, as the team-level quantile search time is inherently non-additive, making it difficult to estimate online, and nontrivial to decompose into consistent individual-level signals for decentralized coordination. To address these challenges, we propose quantile value function factorization (Q-FAC), which learns a set of team-level quantile value functions via distributional temporal-difference methods, factorizes the target quantile value into consistent individual-level signals via a monotonic mixing network, and enables decentralized policy optimization. Extensive experiments on standard MuRS benchmarks, together with validation on a physical multi-robot system, demonstrate the effectiveness and practical feasibility of Q-FAC.

1 Introduction

Multi-robot search (MuRS) concerns coordinating a team of robots to locate a target in a given environment, and constitutes a fundamental problem in robotics and multi-agent systems. MuRS is central to a wide range of real-world applications, including search and rescue [Zhang *et al.*, 2025; Kashyap *et al.*, 2025] and patrolling and surveillance [Cheng *et al.*, 2024], and also serves as a representative testbed for studying fundamental problems in multi-agent coordination, *e.g.*, multi-agent reinforcement learning [Kontogiannis *et al.*, 2025], and distributed control [Dai *et al.*, 2025].

Search performance in MuRS is commonly measured by the time required to locate the target. Accordingly, most existing MuRS formulations adopt *expected* search time as the primary optimization objective. However, in stochastic and dynamic search environments, minimizing the expected search time is often insufficient, as long-tailed or skewed search-time distributions render expectation insensitive to rare but consequential delays. More importantly, such a limitation can fundamentally change the optimal joint search strategy. As illustrated in Fig. 1a, the expectation-optimal and quantile-optimal search strategies can be qualitatively different, even in a simple stationary setting. This discrepancy stems from the inherent non-additivity of quantile-based objectives.



(a) A Simple Scenario

Joint Search Strategy	Expected Search Time	α-quantile Search Time (α=0.9)
R1: 1,2,1,3,4 R2: 1,7,1,5,6	3.4	**4**
R1: 1,3,4,3,1,2 R2: 1,5,6,5,1,7	**2.6**	5

(b) High Quantile VS. Expectation

Figure 1: **A Simple yet Illustrative Example.** (a) Two robots starting from Node 1, cooperatively search for a stationary target located at one of the four bottom nodes with a skewed probability distribution. (b) Expectation-optimal versus α-quantile-optimal (α=0.9) joint search strategies, which yield qualitatively different behaviors.

Motivated by this observation, we formulate a new multi-robot search problem, termed multi-robot *quantile* search (MuRQS), which seeks to minimize a high-percentile search time rather than its expectation. However, directly optimizing a quantile-based search objective in MuRS presents substantial challenges. First, the quantile-based search objective, *i.e.*, the quantile search time, is inherently non-additive, making it difficult to obtain an online recursive estimation form required for learning-based optimization. Second, due to the non-additivity, decomposing a team-level quantile search objective into consistent individual-level signals is nontrivial, as alternative non-additive factorizations often induce local-to-global inconsistency, where improvements at the individual level cannot guarantee improvements in the team-level quantile performance, rendering decentralized policy optimization ineffective.

To address the aforementioned challenges, we propose the quantile value function factorization (Q-FAC) framework, which learns a set of team-level quantile value functions via distributional temporal-difference learning and factorizes the target quantile value into consistent individual-level signals via a monotonic mixing network for decentralized policy optimization. Specifically, Q-FAC comprises three tightly coupled modules. (1) A quantile temporal-difference (QTD) module addresses the non-additivity of quantile value functions by leveraging distributional temporal-difference learning to model the full return distribution, parameterize it with a set of quantile value functions, and thereby enable online estimation of team-level quantiles without constructing a quantile-specific TD recursion. (2) A quantile value function factorization module decomposes the target team-level quantile value function into individual-level quantile signals via a monotonic mixing network, ensuring consistency between individual- and team-level quantile objectives. (3) A quantile policy gradient (QPG) module performs decentralized policy optimization by updating each robot's policy based on the factorized individual-level quantile signals.

The main contributions of this paper are summarized as follows: (1) We formulate a new multi-robot search problem, termed MuRQS, which optimizes high-percentile search time and defines a fundamentally different objective that cannot be reduced to existing expectation-based MuRS formulations. (2) We propose Q-FAC as the first RL-based solution to the MuRQS problem, addressing the challenges of team-level quantile estimation and consistent decentralized coordination through distributional temporal-difference learning and monotonic value factorization. (3) We conduct extensive evaluations on standard MuRS benchmarks and a physical multi-robot system, providing the first systematic empirical validation of quantile-optimal multi-robot search and demonstrating the effectiveness and practical feasibility of Q-FAC.

2 Literature Review

This section provides a brief review of multi-robot search problem taxonomies and mainstream methodologies. In addition, as Q-FAC falls within the broader class of risk-sensitive decision-making, we review related work on risk-sensitive multi-agent reinforcement learning.

2.1 Multi-Robot Search Problem Taxonomies

From the search-objective perspective, existing studies on multi-robot search can be broadly categorized into three representative classes: multi-robot efficient search (MuRES), multi-robot guaranteed search (MuRGS), and multi-robot adversarial search (MuRAS).

MuRES represents the most prevalent and extensively studied class of problems in MuRS. In this setting, the target is typically assumed to be non-adversarial, *i.e.*, its motion dynamics are independent of the robots' search strategies, and the primary objective is to optimize expectation-based performance measures, most commonly by minimizing the expected search time [Ebert *et al.*, 2022; Guo *et al.*, 2025]. Closely related formulations also consider alternative expectation-based criteria, such as maximizing the expected probability of on-time target detection [Asfora *et al.*, 2020; Peng *et al.*, 2025b], or minimizing the expected detection delay [Sheng *et al.*, 2022]. Under the MuRES objective, a wide range of modeling assumptions and solution techniques have been explored, making it the dominant problem formulation in the existing multi-robot search literature.

MuRGS considers a conservative class of multi-robot search problems that aim to coordinate a team of robots to guarantee target detection, regardless of target motion characteristics [Hollinger *et al.*, 2010; Kolling and Kleiner, 2013]. Unlike MuRES, which focuses on optimizing expectation-based performance measures, MuRGS adopts worst-case performance criteria and seeks strategies that guarantee target detection under all admissible target behaviors. As a result, MuRGS is typically studied in settings where strict detection guarantees are required, and robustness against uncertainty in target motion is of primary concern.

MuRAS considers a class of multi-robot search problems in which the target behaves adversarially against the robots' search strategies, actively adapting its motion to evade detection [Rahman *et al.*, 2022; Peng *et al.*, 2025a]. Despite the adversarial target motion, MuRAS typically retains an efficiency-oriented objective and aims to minimize the expected search time, thereby sharing the same expectation-based performance criterion as MuRES. Owing to the adversarial nature of the target dynamics, MuRAS is often studied under game-theoretic frameworks [Fang *et al.*, 2022; Olsen *et al.*, 2022], and differs from MuRES primarily in target motion assumptions rather than the search objective.

This paper introduces a new multi-robot search problem, termed MuRQS, which is characterized by a quantile-based search objective that is fundamentally different from the objectives considered in MuRES, MuRAS, and MuRGS. By adjusting the quantile level, MuRQS can transition from expectation-oriented efficient search to worst-case optimal search, thereby providing an objective-level link between MuRES- and MuRGS-type formulations.

2.2 Methodologies for Multi-Robot Search

To address multi-robot search problems under diverse modeling assumptions and objectives, a wide range of solution methodologies has been developed in the literature. Existing approaches can be categorized into three major classes: optimization-based methods, heuristic-driven approaches, and learning-based methods.

Optimization-based methods constitute the most canonical solutions for the MuRES problem. These approaches formulate multi-robot search as a mathematical optimization problem by explicitly modeling target motion, uncertainty, and robot dynamics, and solve it using off-the-shelf optimization solvers [Asfora *et al.*, 2020; Hollinger *et al.*, 2009]. While offering flexibility in objective and constraint design, these methods typically incur high computational complexity and rely on accurate modeling of the environment, target dynamics, and sensing processes, which limits their applicability to well-structured and well-understood scenarios.

Heuristic-driven approaches address the MuRES problem by prescribing simple local behavior and/or interaction rules for individual robots, from which team-level search behaviors emerge through decentralized execution [Lin *et al.*, 2025; Wang *et al.*, 2025; Ebert *et al.*, 2022]. These methods are intuitive, easy to implement, and highly scalable, making them suitable for large-scale and real-time multi-robot search scenarios. However, due to the lack of an explicit objective-driven formulation, it is often difficult to establish a clear relationship between local rules and the global search objective, limiting their ability to systematically adapt to alternative search objectives.

Learning-based methods, particularly multi-agent reinforcement learning (MARL), model multi-robot search as a sequential decision-making problem and learn decentralized policies through interaction with the environment, typically formulated within the Dec-POMDP framework [Guo *et al.*, 2023a; Wang *et al.*, 2020; Sheng *et al.*, 2022]. These methods eliminate the need for explicit environment and target motion models by learning policies directly from interaction data, enabling adaptive decision-making in dynamic and partially observed search environments. Existing learning-based multi-robot search methods typically optimize expectation-based objectives, and their cooperative training mechanisms are not designed to directly target team-level quantile performance. This gap motivates the development of the Q-FAC framework, an RL-based approach for optimizing high-percentile search time under a quantile-based objective.

2.3 Risk-Sensitive MARL

Risk-sensitive reinforcement learning extends expectation-based objectives by explicitly accounting for outcome variability, tail risk, or worst-case performance. Recent work has further explored risk-sensitive formulations in multi-agent reinforcement learning (MARL), aiming to enable coordinated and risk-aware decision making [Urpí *et al.*, 2021; Jiang *et al.*, 2025; Lim and Malik, 2022; Qiu *et al.*, 2021; Shen *et al.*, 2023]. Representative approaches typically adopt distributional reinforcement learning to model agent-level return distributions, and incorporate risk-sensitive criteria into cooperative learning frameworks through centralized critics or aggregation mechanisms. These studies demonstrate the feasibility of incorporating risk awareness into MARL, but primarily focus on general cooperative decision-making tasks rather than structured multi-robot search problems.

Despite these advances, existing risk-sensitive MARL methods are not directly applicable to the multi-robot quantile search setting considered in this paper. A central limitation is that quantile-based objectives are inherently non-additive, which makes it difficult to define a consistent team-level optimization objective under the decentralized learning scheme. Representative methods such as RiskQ [Shen *et al.*, 2023] approximate the system-level risk by additively aggregating agent-level quantile estimates. However, such additive approximations do not yield a principled formulation for optimizing team-level quantile performance. A more detailed discussion is provided in the supplementary material.

3 Problem Formulation

This section presents a formal formulation of the MuRQS problem. We first describe the problem settings by specifying the environment, target motion dynamics, robots' motion and sensing models, and the quantile-based search objective. We then transform the MuRQS problem into a decentralized partially observable Markov decision process (Dec-POMDP), which provides a principled foundation for the MARL-based approach developed in the next section. A list of major notations used throughout the paper is summarized in Table 1.

Table 1: List of Major Notations Used in the Paper

Notations	**Descriptions**
$\mathcal{G}(\mathcal{V},\mathcal{E})$	the search environment
e_t	target's position at time t
$p_t^{(i)}, o_t^{(i)}$	robot i's position, observation at time t
$\pi^{(i)}$	robot i's decision-making policy
$\boldsymbol{\pi}$	team-level joint decision-making policy
t_{cap}	search time (target's first detection time)
$q_\alpha(X)$	α quantile value of random variable X
N	# of robots
K	# of quantiles
$Z^\pi(s,a)$	team-level return (a random variable)
$Q_\alpha^\pi(s,a)$	team-level α-quantile value function
$Q_\alpha^{(i)}(s,a)$	individual-level α-quantile signal

3.1 Problem Settings

We consider a multi-robot quantile search problem involving a team of N robots and a non-adversarial moving target in a discrete environment, and formalize the problem by defining the system dynamics, sensing and motion models, and the associated quantile-based team-level search objective.

The Environment: The environment is modeled as an undirected, connected unit-cost graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ denotes the set of edges. Both the robots and the target are constrained to occupy nodes in $\mathcal{V}$ and move along edges in $\mathcal{E}$. Under the unit-cost assumption, traversing any edge or remaining at the same node incurs one time step for both the robots and the target. This modeling assumption is commonly adopted in the multi-robot search literature for discrete environments [Guo *et al.*, 2025; Asfora *et al.*, 2020; Sheng *et al.*, 2022]. Non-unit-cost graphs can be equivalently transformed into unit-cost graphs by subdividing edges into multiple unit-cost segments.

Target Motion Dynamics: The target's position at time step t is denoted by $e_t \in \mathcal{V}$ and is unobservable to the robot team. The target is assumed to move non-adversarially, meaning that its motion dynamics are independent of the robots' search strategies and actions. Specifically, the target motion is modeled as a discrete-time Markov process governed by a stochastic transition matrix $\mathbf{\Gamma}$, such that $\mathbb{P}(e_{t+1}|e_t) = \mathbf{\Gamma}(e_t, e_{t+1})$. The transition matrix $\mathbf{\Gamma}$ respects $\mathcal{G}$, allowing the target to either remain at its current node or move to an adjacent node *i.e.*, $e_{t+1} = e_t$ or $(e_t, e_{t+1}) \in \mathcal{E}$. The target motion model $\mathbf{\Gamma}$ is assumed to be unknown to the robot team.

Robot Model: We consider a team of N robots indexed by $i \in \{1,\dots,N\}$, where robot i's position at time step t is denoted by $p_t^{(i)} \in \mathcal{V}$. At each time step, robot i selects an action $a_t^{(i)}$ that moves it to an adjacent node in $\mathcal{G}$ or keeps it at the current node, *i.e.*, $p_{t+1}^{(i)} = p_t^{(i)}$ or $(p_t^{(i)}, p_{t+1}^{(i)}) \in \mathcal{E}$. Each robot is equipped with a local sensor and receives a binary observation $o_t^{(i)} \in \{0,1\}$, where $o_t^{(i)} = 1$ indicates that the target is detected at node $p_t^{(i)}$, and $o_t^{(i)} = 0$ otherwise. The decentralized decision-making policy of robot i, denoted by $\pi^{(i)}$, maps its own history of positions and observations to an action, *i.e.*, $a_t^{(i)} \sim \pi^{(i)}(\cdot \mid p_{\le t}^{(i)}, o_{\le t}^{(i)})$.

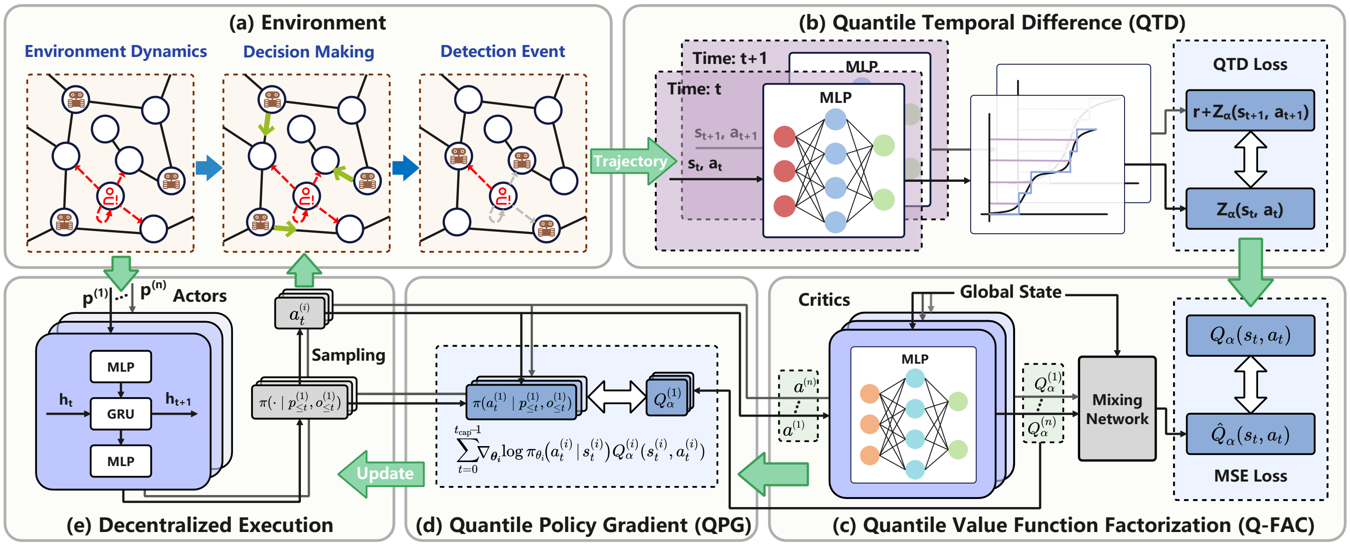


Figure 2: The Q-FAC Framework: (a) **Environment**: interaction between robots and the target until detection; (b) **The QTD Module**: the team-level return distribution is learned via quantile-induced distributional temporal-difference learning, enabling the estimation of team-level quantile value functions; (c) **The Q-FAC Module**: the team-level quantile value is decomposed into individual-level quantile signals through a monotonic mixing network, ensuring consistency between individual- and team-level quantile objectives; (d) **The QPG Module**: individual policies are updated with quantile policy gradient derived from the factorized individual-level quantile signals; (e) **Decentralized Execution**: each robot executes its policy based solely on local observation histories, while coordination is achieved in the centralized training stage.

Search Time and Team-Level Objective: The search process terminates when the target is detected by any robot, and the search time t_{cap} is defined as the first detection time:

$$t_{\text{cap}} \triangleq \inf \left\{ t \geq 0 \;\middle|\; \exists\, i \in \{1, \ldots, N\},\ o_t^{(i)} = 1 \right\}. \tag{1}$$

Due to stochastic target motion and partial observability, t_{cap} is a random variable. The MuRQS objective is to find optimal joint policy $\boldsymbol{\pi}$ that minimizes the α-quantile search time, *i.e.*,

$$q_\alpha(t_{\text{cap}}(\boldsymbol{\pi})) \triangleq \inf \left\{ \tau \in \mathbb{R}^+ \;\middle|\; \mathbb{P}(t_{\text{cap}} \leq \tau) \geq \alpha \right\}, \tag{2}$$

where $0.5 \leq \alpha \leq 1$ specifies the target high-percentile level.

3.2 Transformation into Dec-POMDP

Based on the problem settings in Section 3.1, we recast the MuRQS problem as a decentralized partially observable Markov decision process (Dec-POMDP). The transformation formally bridges the multi-robot search problem and the proposed MARL framework developed in the next section.

Formally, the resulting Dec-POMDP is specified as follows: the agent set is $\mathcal{I} = \{1, \ldots, N\}$; the global state s_t is defined as the joint configuration of the robots and the target, *i.e.*, $s_t = (p_t^{(1)}, \ldots, p_t^{(N)}, e_t) \in \mathcal{S}$; each robot i selects an action $a_t^{(i)} \in \mathcal{A}^{(i)}(p_t^{(i)})$ according to the individual policy $\pi^{(i)}$, and receives a local observation $o_t^{(i)} \in \{0, 1\}$; the state transition function is induced by the robots' actions and the target transition matrix $\boldsymbol{\Gamma}$; the episode terminates upon target detection. We define a per-step reward $r_t = 1$ until termination. Under the episodic and undiscounted setting, the cumulative return equals the search time in MuRQS, *i.e.*,

$$Z^{\boldsymbol{\pi}}(s_0) = t_{\text{cap}}, \tag{3}$$

which aligns the Dec-POMDP return with the quantile-based search objective defined in Section 3.1.

4 The Q-FAC Framework

This section presents the proposed quantile value function factorization (Q-FAC) framework for solving the multi-robot quantile search (MuRQS) problem. Due to the non-additivity of quantile-based objectives and the absence of a recursive temporal-difference formulation, Q-FAC adopts an indirect yet principled approach that models the team-level return distribution and induces corresponding quantile value functions from the learned distribution, enabling consistent factorization and decentralized policy optimization. An overview of the Q-FAC framework is shown in Fig. 2, and its components are detailed in the following subsections.

4.1 Quantile Temporal Difference (QTD)

The direct TD estimation of quantile value functions is inapplicable due to their non-additivity and the lack of a Bellman-style recursion. We therefore perform recursion at the distribution level by learning the team-level return distribution via distributional temporal-difference updates. The return distribution is represented by a finite set of team-level quantile value functions, from which the target quantile value function can be directly obtained. Let $Z^{\boldsymbol{\pi}}(s, a)$ denote the random team-level return obtained by executing joint action a at state s and thereafter following the joint policy $\boldsymbol{\pi}$. The α-quantile value of $Z^{\boldsymbol{\pi}}(s, a)$ is denoted by $Q_\alpha^{\boldsymbol{\pi}}(s, a)$.

Definition 4.1 (The Distributional Bellman Equation)**.** *Under the undiscounted episodic setting, the return random variable $Z^{\boldsymbol{\pi}}(s, a)$ satisfies the following distributional Bellman equation:*

$$Z^{\boldsymbol{\pi}}(s, a) \overset{D}{=} r(s, a) + Z^{\boldsymbol{\pi}}(s', a'), \tag{4}$$

where $\overset{D}{=}$ denotes equality in distribution, s' is the next state *induced by (s,a), and $a' \sim \boldsymbol{\pi}(\cdot \mid s')$.*

Remark 4.1. *In the context of MuRQS, the distributional Bellman equation is essential to enable recursive learning under quantile-based search objectives; our contribution lies not in the equation itself, but in exploiting the distribution-level recursion to induce the team-level quantile value function and subsequently facilitate consistent quantile function factorization despite the non-additivity.*

Note that both sides of Eq. (4) are random variables, and we measure the discrepancy between their induced distributions using the Wasserstein distance.

Definition 4.2 (Wasserstein Distance)**.** *For two one-dimensional distributions Z_1 and Z_2 with cumulative distribution functions F_{Z_1} and F_{Z_2}, the p-Wasserstein distance is defined as:*

$$W_p(Z_1, Z_2) = \left(\int_0^1 \left| F_{Z_1}^{-1}(\tau) - F_{Z_2}^{-1}(\tau) \right|^p \, d\tau \right)^{\frac{1}{p}}, \quad (5)$$

where $F_Z^{-1}(\tau)$ denotes the quantile function of Z. For one-dimensional distributions, the Wasserstein distance admits a closed-form expression in terms of quantile functions.

Based on the distributional Bellman equation in Definition 4.1, we construct a sample-based distributional temporal-difference update by minimizing the Wasserstein distance between the predicted return distribution and the target distribution. Specifically, the distributional TD update minimizes the following objective:

$$\mathcal{L}_{\text{DTD}}(\boldsymbol{w}) = W_p(Z_{\boldsymbol{w}}(s,a),\ r(s,a) + Z_{\boldsymbol{w}^-}(s',a')), \quad (6)$$

where $Z_{\boldsymbol{w}}$ denotes the parameterized return distribution with parameter vector $\boldsymbol{w}$. The second term is treated as a distributional TD target, parameterized by a target network with parameters $\boldsymbol{w}^-$.

In practice, directly optimizing the continuous quantile function is intractable. We therefore approximate the return distribution using a finite set of quantile value functions.

Definition 4.3 (Finite Quantile Approximation)**.** *We approximate the return distribution $Z_{\boldsymbol{w}}(s,a)$ by a* finite *set of K quantile value functions $\{Q_{\alpha_j}(s,a;\boldsymbol{w})\}_{j=1}^K$, where each $\alpha_j \in (0,1)$ denotes a predefined quantile level. Specifically, the return distribution is represented as*

$$Z_{\boldsymbol{w}}(s,a) \overset{D}{\approx} Q_{\boldsymbol{w}}(s,a;\alpha), \quad \alpha \sim \mathit{Unif}\{\alpha_j\}_{j=1}^K.$$

With the finite quantile approximation, we denote the bootstrapped target random variable by Z', *i.e.*, $Z'(s,a) = r(s,a) + Z_{\boldsymbol{w}^-}(s',a')$. Correspondingly, the α-quantile value of Z' is denoted by $Q'_\alpha(s,a)$. The quantile induced distributional TD loss is then given by:

$$\mathcal{L}_{\text{QTD}}(\boldsymbol{w}) = \left(\frac{1}{K} \sum_{j=1}^{K} \left| Q_{\alpha_j}(s,a;\boldsymbol{w}) - Q'_{\alpha_j}(s,a;\boldsymbol{w}^-) \right|^p \right)^{\frac{1}{p}}, \quad (7)$$

where $\boldsymbol{w}^-$ are the target network's parameters that are periodically copied from $\boldsymbol{w}$ and kept constant for several iterations.

Remark 4.2. *The above quantile-induced TD loss provides a consistent empirical approximation of the Wasserstein distance between return distributions under the finite quantile representation scheme.*

4.2 Quantile Function Factorization (Q-FAC)

The QTD module in Section 4.1 enables the learning of team-level quantile value functions. However, MuRQS is inherently a multi-agent problem, and directly optimizing a centralized team-level quantile value function is insufficient for decentralized execution. Unlike expectation-based value functions, quantile value functions are non-additive, rendering summation-based decompositions invalid, while naive non-additive factorizations may induce local-to-global inconsistency. To address these challenges, we introduce a monotonic quantile value function factorization that decomposes the team-level quantile value into individual-level quantile signals via a monotonic mixing network, ensuring consistency between individual- and team-level quantile objectives.

We consider a monotonic mixing architecture to decompose the team-level quantile value function into individual-level quantile utilities. Specifically, for a given quantile level $\alpha \in (0,1)$, the team-level quantile value function is expressed as:

$$Q_\alpha^{\boldsymbol{\pi}}(s,\boldsymbol{a}) = f_\alpha\Big(Q_\alpha^{(1)}(s_1,a_1),\ \ldots,\ Q_\alpha^{(N)}(s_N,a_N);\ s\Big), \quad (8)$$

where $Q_\alpha^{(i)}(s_i,a_i)$ denotes the individual-level α-quantile signal of robot i, and $f_\alpha(\cdot)$ is a mixing network whose parameters are generated by a state-conditioned hyper-network. To ensure consistency between individual- and team-level quantile objectives, the mixing network is constrained to be monotonic with respect to each individual quantile utility, *i.e.*,

$$\frac{\partial Q_\alpha^{\boldsymbol{\pi}}(s,\boldsymbol{a})}{\partial Q_\alpha^{(i)}(s_i,a_i)} \geq 0, \qquad \forall i \in \mathcal{I},\ \forall \alpha \in (0,1). \quad (9)$$

This constraint is enforced by restricting the hyper-network to generate non-negative mixing weights, and we present the Q-IGM theorem whose proof is in the supplementary material.

Theorem 1 (Quantile Individual Global Minimum (Q-IGM))**.** *Suppose that, for a fixed quantile level α, the team-level quantile value function $Q_\alpha^{\boldsymbol{\pi}}(s,\boldsymbol{a})$ admits a monotonic factorization with respect to the individual quantile utilities $\{Q_\alpha^{(i)}(s_i,a_i)\}_{i=1}^N$ satisfying Eq. (9). Then, the greedy minimization of the team-level α-quantile objective is* consistent *with the greedy minimization of individual-level α-quantile objectives, i.e.,*

$$\arg\min_{\boldsymbol{a}} Q_\alpha^{\boldsymbol{\pi}}(s,\boldsymbol{a}) = \left(\arg\min_{a_i} Q_\alpha^{(i)}(s_i,a_i) \right)_{i=1}^N. \quad (10)$$

Remark 4.3. *Theorem 1 establishes that monotonicity is a sufficient condition for resolving the non-additivity of quantile value functions and avoiding local-to-global inconsistency in decentralized optimization.*

4.3 Quantile Policy Gradient (QPG)

Based on the monotonic quantile value function factorization and the Q-IGM theorem established in Section 4.2, decentralized policy optimization can be performed by minimizing individual-level quantile signals. Specifically, since the greedy minimization of the team-level α-quantile objective is

Algorithm 1 Q-FAC's Training Process

Input: (1) Graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$; (2) Target quantile level α; (3) # of Robots N; (4) Max. # of Episodes: $E_{\max}$.
Output: Decentralized policies parameters $\{\theta_i\}_{i\in\mathcal{I}}$
Initialize:
(1) Individual policy network parameters; $\{\theta_i\}_{i\in\mathcal{I}}$;
(2) Individual quantile network parameters $\{w_i\}_{i\in\mathcal{I}}$;
(3) Mixing network parameters $\boldsymbol{\phi}$;
(4) Team quantile network parameters $\boldsymbol{w}$;
(5) Robot team's initial position $p_0^{(i)}$;

1: Replay buffer $\mathcal{D} = \{\}$, episode $\leftarrow 0$;
2: **while** episode $\leq E_{\max}$ **do**
3: $\quad t \leftarrow 0$, $\boldsymbol{w}^- \leftarrow \boldsymbol{w}$;
4: $\quad \forall i \in \mathcal{I} : o_t^{(i)} \leftarrow 0,\ p_t^{(i)} \leftarrow p_0^{(i)}$;
5: $\quad$ **while** $\forall i \in \mathcal{I} : o_t^{(i)} = 0$ **do**
6: $\quad\quad$ Each robot i selects action $a_t^{(i)} \sim \pi^{(i)}(\cdot \mid p_{\leq t}^{(i)}, o_{\leq t}^{(i)})$;
7: $\quad\quad$ Execute $a_t^{(i)}$, transition to $p_{t+1}^{(i)}$, observe $o_{t+1}^{(i)}$;
8: $\quad\quad$ Observe team reward $r_t \leftarrow 1$;
9: $\quad\quad$ Store $\{p_t^{(i)}, o_t^{(i)}, a_t^{(i)}, p_{t+1}^{(i)}, o_{t+1}^{(i)}\}_{i\in\mathcal{I}}$ and r_t into $\mathcal{D}$; $t \leftarrow t+1$;
10: $\quad$ **end while**
11: $\quad$ Sample a mini-batch of transitions from $\mathcal{D}$;
12: $\quad$ Compute $\mathcal{L}_{\mathrm{QTD}}$ in Eq. (7) to update $\{w_i\}_{i\in\mathcal{I}}$, $\boldsymbol{\phi}$, $\boldsymbol{w}$;
13: $\quad$ Compute QPG in Eq. (11) to update $\{\theta_i\}_{i\in\mathcal{I}}$;
14: $\quad$ episode $\leftarrow$ episode + 1.
15: **end while**

consistent with that of the individual-level α-quantile objectives, each robot can update its policy with a local quantile policy gradient via the stochastic gradient *descent* method. The resulting quantile policy gradient (QPG) information for robot i is given by:

$$\nabla_{\boldsymbol{\theta}} J_\alpha^{(i)}(\boldsymbol{\theta}_i) \approx \sum_{t=0}^{t_{\mathrm{cap}}-1} \nabla_{\boldsymbol{\theta}_i} \log \pi_{\theta_i}\big(a_t^{(i)} \,|\, s_t^{(i)}\big) Q_\alpha^{(i)}(s_t^{(i)}, a_t^{(i)}), \quad (11)$$

where θ_i denotes the parameters of agent i's decision-making policy π_i. Note that for notational simplicity, we use $s_t^{(i)}$ as π_i's decision-making basis; in practice, π_i will take as inputs robot i's history of positions and observations, *i.e.*, $p_{\leq t}^{(i)}$ and $o_{\leq t}^{(i)}$. The right hand side of Eq. (11) is a stochastic approximation of the true gradient, thus we use the symbol '$\approx$'. The rationale that Eq. (11) aligns with the gradient of team-level quantile value function is presented in the Appendix.

4.4 Computational Complexity Analysis

Algorithm 1 presents the pseudocode of Q-FAC, and we employ the Big-$\mathcal{O}$ notation [Knuth, 1976] to analyze the computational complexity of the training process of Q-FAC. For simplicity, all neural network layers are assumed to have a uniform hidden dimension d. The main computational cost arises within the episode loop (lines 2–14), which can be divided into two parts: environment interaction and network updates. During interaction, N robots select actions via their policy networks and generate trajectories of average length $\bar{T}$, leading to the computational complexity of $\mathcal{O}(\bar{T}Nd^2)$. In the update phase, let B denote the batch size and K the number of quantiles. Computing the QTD loss (line 11) requires $\mathcal{O}(B(d^2+K^2))$, while the QPG update (line 12) costs $\mathcal{O}(BNd^2)$. By aggregating these and omitting non-leading terms, the overall time complexity can be expressed as:

$$\mathcal{O}\big((\bar{T}+B)E_{\max}N(d^2+K^2)\big).$$

Therefore, Q-FAC's training process exhibits polynomial complexity with respect to $(E_{\max}, N, d, K, B)$, and in particular scales linearly with the number of robots N.

5 Simulation Results and Analysis

This section evaluates Q-FAC on standard multi-robot search (MuRS) benchmarks by comparing its high-quantile search performance with state-of-the-art baselines, and further conducts ablation studies to examine the contributions of each component within Q-FAC.

Specifically, the baseline algorithms, summarized in Table 2, span three representative categories: prevailing MuRS methods, canonical MARL algorithms, and risk-sensitive approaches. All baselines are adapted to the MuRQS setting with identical observation and action spaces to ensure fair comparison. Detailed meta-parameter configurations and additional experimental results are provided in the supplementary material, and the source code is publicly available[1].

Table 2: Summary of Selected Baseline Algorithms

Category	Methodology
MuRS Methods	MILP [Asfora *et al.*, 2020] DRL-Searcher [Guo *et al.*, 2023b] PD-FAC [Sheng *et al.*, 2022]
MARL Algorithms	VDN [Sunehag *et al.*, 2018] QMIX [Rashid *et al.*, 2020] MAPPO [Yu *et al.*, 2022]
Risk-Sensitive Algs.	RiskQ [Shen *et al.*, 2023] D-FAC [Sun *et al.*, 2021]

5.1 Benchmark Comparison

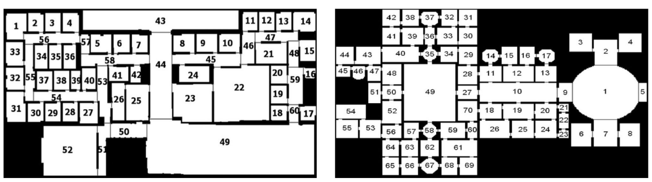
We evaluate Q-FAC against state-of-the-art baselines on two standard MuRS benchmarks, OFFICE and MUSEUM (Fig. 4). In both environments, robots start from fixed initial nodes, while the target is randomly initialized over the environment. At each time step, it moves non-adversarially to one of its adjacent nodes with equal probability. An episode terminates once any robot occupies the same node as the target.

Table 3 reports the high-quantile performance comparison between Q-FAC and state-of-the-art baselines across different team sizes ($N \in \{2,3,4,5\}$) and quantile levels ($\alpha \in \{0.90, 0.95\}$). From the results, we observe that (1) Q-FAC consistently achieves the best or second-best α-quantile performance across all configurations, demonstrating its robustness to variations in team size and quantile level; (2) The performance advantage of Q-FAC becomes increasingly pronounced as α increases, under which many baselines frequently fail to locate the target within the time limit,

[1] https://anonymous.4open.science/r/Q-FAC-8VE6rK

Table 3: High-quantile performance comparison between Q-FAC and baseline algorithms (the detection time is capped at 300 time steps). **Bold** numbers indicate the best α-quantile performance, and <u>underlined</u> numbers indicate the second-best performance.

Network	α	N	Q-FAC	VDN	QMIX	DRL-Searcher	PD-FAC	MAPPO	RiskQ	MILP	D-FAC
OFFICE	0.90	2	**29.2**	<u>56.4</u>	68.2	300	118.5	95.3	300	71.1	300
		3	**23**	52.6	<u>24.1</u>	180	114.4	77.1	300	44	157.4
		4	**22.1**	43.9	<u>23.1</u>	168.4	26	159	34.8	30.1	140.6
		5	**20.1**	26.2	23	119	21.3	49.1	<u>21</u>	27.2	73.1
	0.95	2	**77.7**	164.2	250.7	300	261.4	180.2	300	<u>81.1</u>	300
		3	**37**	212.4	61.5	283.4	184.5	167.1	300	<u>48.3</u>	300
		4	<u>34.1</u>	106.4	**31**	235.2	52.3	235.8	100.8	35.4	300
		5	**23.1**	34.1	<u>27.2</u>	171.3	34.9	65.1	29.1	32.2	123.6
MUSEUM	0.90	2	**132.8**	208.9	300	239.6	300	236.6	300	<u>147</u>	300
		3	**75.8**	152.9	120.2	121.2	296.4	<u>118</u>	217.9	127.8	300
		4	**59.7**	188.7	<u>63.2</u>	94.1	283.8	115.9	211.9	66.4	300
		5	**34.5**	153.2	<u>36.3</u>	81.7	141.3	85.1	185.6	62.3	300
	0.95	2	**190.2**	300	300	300	300	300	300	<u>208</u>	300
		3	**131**	284.8	246.3	<u>147.1</u>	300	181.9	296.6	147.2	300
		4	**102.4**	294.3	130.1	<u>117.4</u>	300	160.9	292	132	300
		5	<u>78.1</u>	222.1	**61.2**	97.1	236.5	134.2	265.5	122.6	300



(a) OFFICE (b) MUSEUM

Figure 3: Canonical MuRS benchmarks from [Hollinger *et al.*, 2009], each room is associated with a corresponding node number.

resulting in saturated quantile values (300); and (3) Risk-sensitive baselines such as RiskQ do not consistently outperform expectation-based baselines, indicating that modeling individual-level risk alone is insufficient for optimizing team-level high-quantile search performance. These results highlight the necessity of explicitly learning and factorizing team-level quantile value functions, as achieved by Q-FAC.

5.2 Ablation Study

This subsection conducts an ablation study to isolate the contribution of each core module in Q-FAC. Specifically, we compare three ablated variants against the full Q-FAC framework which comprises QTD estimation, Q-FAC factorization, and QPG optimization. (1) **TD-FAC**, which replaces the QTD module with the canonical TD learner while retaining the Q-FAC and QPG modules; (2) **Q-VDN**, which replaces the Q-FAC module with a VDN-style additive decomposition while keeping the QTD and QPG modules unchanged; and (3) **Q-FAC(ϵ)**, which preserves both QTD and Q-FAC modules but replaces the QPG module with a simple ϵ-greedy policy.

Table 4 shows that the full Q-FAC consistently achieves the best overall performance across both benchmarks. TD-FAC regresses to QMIX, as replacing QTD with canonical TD effectively reduces the framework to an expectation-based method, failing to capture the tail behavior. Q-VDN also underperforms, since directly summing individual quantile functions violates the non-additivity of quantiles and leads to inconsistent team-level behavior. Although Q-FAC(ϵ) is structurally closest to Q-FAC, the ϵ-greedy exploration behavior conflicts with the tail-sensitive nature of the quantile objective, resulting in the degraded performance.

Table 4: Ablation study on the α-quantile search time ($\alpha = 0.9$).

Network	N	TD-FAC	Q-VDN	Q-FAC(ϵ)	Q-FAC
OFFICE	2	<u>68.2</u>	207.1	208.4	**29.2**
	3	<u>24.1</u>	132.2	98.76	**23**
	4	<u>23.1</u>	121.6	37.6	**22.1**
	5	<u>23</u>	96.4	24.9	**20.1**
MUSEUM	2	300	<u>259.1</u>	300	**132.8**
	3	<u>120.2</u>	228.1	300	**75.8**
	4	<u>63.2</u>	182.5	261.8	**59.7**
	5	<u>36.3</u>	151.3	251.4	**34.5**

6 Conclusion

This paper introduces a new multi-robot search problem, termed MuRQS, which targets minimizing high-percentile search time. Accordingly, we propose Q-FAC, the first MARL-based solution to the MuRQS problem, following an ‘estimation–factorization–optimization’ procedure that combines QTD-based distributional estimation, quantile-consistent value factorization, and quantile policy gradient optimization. We evaluate Q-FAC against several baseline algorithms on standard MuRS benchmarks, and further conduct ablation studies to validate the contribution of each module.

Future work will investigate quantile-consistent value function factorization as a general design principle for decentralized risk-sensitive MARL, including its applicability to other cooperative tasks beyond multi-robot search.

References

[Asfora *et al.*, 2020] Beatriz A Asfora, Jacopo Banfi, and Mark Campbell. Mixed-integer linear programming models for multi-robot non-adversarial search. *IEEE Robotics and Automation Letters (RA-L)*, 5(4):6805–6812, 2020.

[Cheng *et al.*, 2024] Hao Cheng, Jin Yi, Wei Xia, Huayan Pu, and Jun Luo. Adaptive memetic algorithm with dual-level local search for cooperative route planning of multi-robot surveillance systems. *Complex System Modeling and Simulation*, 4(2):210–221, 2024.

[Dai *et al.*, 2025] Pengcheng Dai, Yuanqiu Mo, Wenwu Yu, and Wei Ren. Distributed neural policy gradient algorithm for global convergence of networked multiagent reinforcement learning. *IEEE Transactions on Automatic Control (TAC)*, 70(11):7109–7124, 2025.

[Ebert *et al.*, 2022] Julia T. Ebert, Florian Berlinger, Bahar Haghighat, and Radhika Nagpal. A hybrid PSO algorithm for multi-robot target search and decision awareness. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 11520–11527, 2022.

[Fang *et al.*, 2022] Xu Fang, Chen Wang, Lihua Xie, and Jie Chen. Cooperative pursuit with multi-pursuer and one faster free-moving evader. *IEEE Transactions on Cybernetics (ToC)*, 52(3):1405–1414, 2022.

[Guo *et al.*, 2023a] Hongliang Guo, Zhaokai Liu, Rui Shi, Wei-Yun Yau, and Daniela Rus. Cross-entropy regularized policy gradient for multirobot nonadversarial moving target search. *IEEE Transactions on Robotics (T-RO)*, 39(4):2569–2584, 2023.

[Guo *et al.*, 2023b] Hongliang Guo, Qihang Peng, Zhiguang Cao, and Yaochu Jin. DRL-Searcher: A unified approach to multirobot efficient search for a moving target. *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 35(3):3215–3228, 2023.

[Guo *et al.*, 2025] Hongliang Guo, Qi Kang, Wei-Yun Yau, Chee-Meng Chew, and Daniela Rus. R-FAC: Resilient value function factorization for multirobot efficient search with individual failure probabilities. *IEEE Transactions on Robotics (T-RO)*, 41:3385–3401, 2025.

[Hollinger *et al.*, 2009] Geoffrey Hollinger, Sanjiv Singh, Joseph Djugash, and Athanasios Kehagias. Efficient multi-robot search for a moving target. *The International Journal of Robotics Research (IJRR)*, 28(2):201–219, 2009.

[Hollinger *et al.*, 2010] Geoffrey Hollinger, Athanasios Kehagias, and Sanjiv Singh. GSST: Anytime guaranteed search. *Autonomous Robots (AR)*, 29(1):99–118, 2010.

[Jiang *et al.*, 2025] Jiahui Jiang, Wang He, and Wenwu Yu. Risk-averse multi-agent reinforcement learning with distributional mean–variance formulation. In *International Conference on Artificial Intelligence and Machine Learning Research (CAIMLR)*, volume 13635. SPIE, 2025.

[Kashyap *et al.*, 2025] Gautam Siddharth Kashyap, Deepkashi Mahajan, Orchid Chetia Phukan, Ankit Kumar, Alexander EI Brownlee, and Jiechao Gao. From simulations to reality: enhancing multi-robot exploration for urban search and rescue. *International Journal of Information Technology (IJIT)*, pages 1–12, 2025.

[Knuth, 1976] Donald E Knuth. Big Omicron and big Omega and big Theta. *ACM Sigact News*, 8(2):18–24, 1976.

[Kolling and Kleiner, 2013] Andreas Kolling and Alexander Kleiner. Multi-UAV motion planning for guaranteed search. In *Proceedings of the 12th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, AAMAS ’13, page 79–86, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[Kontogiannis *et al.*, 2025] Andreas Kontogiannis, Konstantinos Papathanasiou, Yi Shen, Giorgos Stamou, Michael M. Zavlanos, and George Vouros. Enhancing cooperative multi-agent reinforcement learning with state modelling and adversarial exploration. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267, pages 31437–31466. PMLR, 13–19 Jul 2025.

[Lim and Malik, 2022] Shiau Hong Lim and Ilyas Malik. Distributional reinforcement learning for risk-sensitive policies. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 30977–30989, 2022.

[Lin *et al.*, 2025] Xiankun Lin, Feng Gao, and Wenhui Bian. A high-effective swarm intelligence-based multi-robot cooperation method for target searching in unknown hazardous environments. *Expert Systems with Applications (ESWA)*, 262:125609, 2025.

[Olsen *et al.*, 2022] Trevor Olsen, Nicholas M. Stiffler, and Jason M. O’Kane. Robust-by-design plans for multi-robot pursuit-evasion. In *International Conference on Robotics and Automation (ICRA)*, pages 10716–10722, 2022.

[Peng *et al.*, 2025a] Qihang Peng, Hongliang Guo, Boyang Li, Chih-Yung Wen, and Yaochu Jin. SMC-Searcher: Signal mediated coordination for decentralized multi-robot adversarial moving target search. *IEEE Transactions on Emerging Topics in Computational Intelligence (T-ETCI)*, 9(5):3399–3412, 2025.

[Peng *et al.*, 2025b] Qihang Peng, Hongliang Guo, Zhengyan Zhang, Chih-Yung Wen, and Yaochu Jin. PF-MAAC: A learning-based method for probabilistic optimization in time-constrained non-adversarial moving target search. *Swarm and Evolutionary Computation (SEC)*, 92:101785, 2025.

[Qiu *et al.*, 2021] Wei Qiu, Xinrun Wang, Runsheng Yu, Rundong Wang, Xu He, Bo An, Svetlana Obraztsova, and Zinovi Rabinovich. RMIX: Learning risk-sensitive policies for cooperative reinforcement learning agents. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 23049–23062, 2021.

[Rahman *et al.*, 2022] A. Azizur Rahman, Arnab Bhattacharya, Thiagarajan Ramachandran, Sayak Mukherjee, Himanshu Sharma, Ted Fujimoto, and Samrat Chatterjee. AdverSAR: Adversarial search and rescue via multi-agent reinforcement learning. In *2022 IEEE International*

Symposium on Technologies for Homeland Security (HST), pages 1–7, 2022.

[Rashid *et al.*, 2020] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research (JMLR)*, 21(178):1–51, 2020.

[Shen *et al.*, 2023] Siqi Shen, Chennan Ma, Chao Li, Weiquan Liu, Yongquan Fu, Songzhu Mei, Xinwang Liu, and Cheng Wang. RiskQ: risk-sensitive multi-agent reinforcement learning value factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 34791–34825, 2023.

[Sheng *et al.*, 2022] Wenda Sheng, Hongliang Guo, Wei-Yun Yau, and Yingjie Zhou. PD-FAC: Probability density factorized multi-agent distributional reinforcement learning for multi-robot reliable search. *IEEE Robotics and Automation Letters (RA-L)*, 7(4):8869–8876, 2022.

[Sun *et al.*, 2021] Wei-Fang Sun, Cheng-Kuang Lee, and Chun-Yi Lee. DFAC framework: Factorizing the value function via quantile mixture for multi-agent distributional Q-learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 9945–9954. PMLR, 2021.

[Sunehag *et al.*, 2018] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning. In *Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 2085–2087, 2018.

[Urpí *et al.*, 2021] Núria Armengol Urpí, Sebastian Curi, and Andreas Krause. Risk-averse offline reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2021.

[Wang *et al.*, 2020] Yuanda Wang, Lu Dong, and Changyin Sun. Cooperative control for multi-player pursuit-evasion games with reinforcement learning. *Neurocomputing*, 412:101–114, 2020.

[Wang *et al.*, 2025] Miao Wang, Bin Xin, Mengjie Jing, and Yun Qu. A priority-based multi-robot search algorithm for indoor source searching. *IEEE Transactions on Automation Science and Engineering (TASE)*, 2025.

[Yu *et al.*, 2022] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 24611–24624, 2022.

[Zhang *et al.*, 2025] Qilong Zhang, Yi Feng, and Cheng Li. Post-disaster multi-robot target search based on improved distributed reinforcement learning. In *2025 44th Chinese Control Conference (CCC)*, pages 1–6. IEEE, 2025.