Learning-Based Maximum Likelihood Estimator for Angle-of-Arrival Localization

Chengyi Zhou, Meiqin Liu, *Senior Member, IEEE*, Senlin Zhang, *Member, IEEE*, Ronghao Zheng, *Member, IEEE*, Shanling Dong, *Member, IEEE*, and Zhunga Liu, *Member, IEEE*

***Abstract*—The estimation of target positions from angle-of-arrival (AOA) measurements has been extensively researched, and various estimators have been proposed to tackle this challenge. Among these, the maximum likelihood estimator (MLE) is notable for its well-recognized properties, including asymptotic unbiasedness and efficiency. However, traditional MLEs, such as the Gauss-Newton algorithm, often encounter difficulties due to the need for a first-order linearization step in computing the Jacobian matrix. This requirement introduces the potential for significant errors and convergence issues, especially in highly nonlinear systems. To overcome this limitation, this paper introduces a learning framework to address the maximum likelihood estimation problem, where the iterative increments are treated as the output of the agent's actions. Building upon this framework, we develop a learning-based MLE. Comprehensive numerical simulation results demonstrate the effectiveness and superiority of our approach. First, it effectively resolves convergence issues associated with linearization in traditional MLEs. Second, it exhibits robust adaptability by successfully solving both two-dimensional and three-dimensional AOA localization problems. Last, the proposed method significantly enhances localization accuracy compared to existing estimators.**

***Index Terms*—Angle-of-arrival localization, maximum likelihood estimator, Gauss-Newton algorithm, first-order linearization, deep reinforcement learning.**

Received 28 December 2023; revised 4 June 2024; accepted 11 July 2024. Date of publication 30 July 2024; date of current version 3 December 2024. This work was supported in part by Zhejiang Provincial Natural Science Foundation of China under Grant LZ23F030006, in part by the National Natural Science Foundation of China under Grant 62173299 and Grant U23B2060, in part by the Joint Fund of Ministry of Education for Pre-research of Equipment under Grant 8091B022147, Grant 8091B032234, and Grant 8091B042220, and in part by the Fundamental Research Funds for Xi'an Jiaotong University under Grant xtr072022001. The associate editor coordinating the review of this article and approving it for publication was Dr. Yuxin Chen. *(Corresponding author: Meiqin Liu.)*

Chengyi Zhou, Senlin Zhang, Ronghao Zheng, and Shanling Dong are with the College of Electrical Engineering and the National Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou 310027, China (e-mail: chengyizhou@zju.edu.cn; slzhang@zju.edu.cn; rzheng@zju.edu.cn; shanlingdong28@zju.edu.cn).

Meiqin Liu is with the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Xi'an Jiaotong University, Xi'an 710049, China, and also with the College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: liumeiqin@zju.edu.cn).

Zhunga Liu is with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: liuzhunga@nwpu.edu.cn).

Digital Object Identifier 10.1109/TSP.2024.3434979

I. Introduction

ANGLE-OF-ARRIVAL (AOA) localization has garnered sustained attention over several decades due to its wide range of applications in both civilian and military domains, including sonar, radar, electronic warfare, surveillance, and navigation [1], [2], [3], [4], [5], [6]. AOA localization is a nonlinear state estimation problem that aims to determine the unknown position of a target based on noise-corrupted bearing measurements collected by a moving observer or multiple spatially distributed sensors. The data available for AOA localization consists of a series of noise-corrupted angles from observation sensors to the target. On the signal processing opinion, the highly nonlinear relationship between the bearing measurements and the target position makes the AOA localization a nontrivial task.

Over the years, researchers have developed numerous approaches to solve the AOA localization problem. These solutions can be broadly categorized into two groups: closed-form solutions and iterative search solutions. One of the most well-known closed-form solutions is the Stansfield estimator [7], which utilizes a linearized least-squares procedure to determine the position of a target. However, this estimator assumes that the range information between the sensors and the target is available. This strong assumption can be avoided by employing the method of orthogonal vectors (OV), which transforms the Stansfield estimator into the pseudolinear estimator (PLE) [16]. The PLE method has merits of simplicity and low computational requirements, but it is susceptible to significant bias issues resulting from the correlation between the measurement matrix and the bearing noise [9].

Several approaches have been developed to mitigate the bias problem in PLE. In [10], a recursive instrumental variable estimator was proposed, utilizing target state estimates from past measurements to construct an instrumental variable matrix. In [11], a non-iterative weighted instrumental variable estimator (WIVE) was designed, leveraging instrumental variables obtained from a biased location estimate to eliminate the correlation between the measurement matrix and the pseudolinear noise vector. The application of total least squares (TLS) to AOA localization was considered in [12] to alleviate the bias problem, where the bias is reduced by considering errors in both the system matrix and the data. The aforementioned studies provide closed-form solutions for two-dimensional (2D) AOA localization, which cannot be directly extended to the three-

dimensional (3D) space. In response to this limitation, [13] proposed the orthogonal vector estimator (OVE) to address the 3D AOA localization problem, which is essentially identical to the PLE in the 2D space. Furthermore, a 3D-PLE was formulated by projecting the bearing lines onto the xy-plane and subsequently applying the PLE to the resulting 2D localization problem. However, OVE and 3D-PLE also produce biased position estimates. Therefore, the 3D-WIVE was designed to address this issue. In [14], a 3D bias reduction solution was presented by expanding the parameter space, recasting the pseudolinear equations in augmented form, and imposing a quadratic constraint on the unknowns. [15] extended the Stansfield estimator to 3D and proposed two algorithms, named WS3D and IVWS3D, which utilize estimated range and bearing quality weightings to enhance the performance.

The iterative search solution determines the target position based on the maximum likelihood (ML) estimation criterion [16], [17], [18], [19], [20]. In [21], a detailed examination of the bias and variance analysis of MLE reveals that it enjoys desirable properties such as asymptotic unbiasedness and efficiency. Generally, an MLE can be developed by solving the optimization problem of the form $\boldsymbol{X}^* = \arg\min \mathcal{F}(\boldsymbol{X})$, where $\mathcal{F}$ denotes the negative log-likelihood function, $\boldsymbol{X}$ stands for the decision variable, and $\boldsymbol{X}^*$ represents the optimized decision. Note that under the assumption of normal density, the optimization problem is equivalent to a nonlinear least squares problem. One solution to this problem is to utilize the Gauss-Newton (GN) algorithm. The GN algorithm relies on linearizing the system around the current estimate and updating the parameters based on the linearized approximation [12], [18]. However, in highly nonlinear systems, the linearization may not accurately capture the behavior of the system, leading to large errors or divergence in the estimation process. This is because the local linearity assumption becomes less valid as the nonlinearity of the system increases. Hence, the design of an accurate optimization algorithm that can relax the linearization requirement is important.

Recently, deep reinforcement learning (DRL) has found successful applications in various radar signal processing (RSP) tasks [22], [23], [24], [25], [26]. For instance, Thornton et al. introduced a DRL framework for a spectrum-sharing cognitive radar system [27]. Their work shows that the DRL approach not only improves detection performance compared to the sense-and-avoid method but also ensures stability in the presence of both stationary and non-stationary stochastic interference. To tackle the challenge of low training efficiency due to limited channel state information in [27], a DRL-based cognitive multicarrier radar (CMCR) for communication interference avoidance was proposed in [28]. Notably, the DRL-based CMCR demonstrates remarkable learning and detection capabilities compared to traditional methods. In [29], Jiang et al. employed DRL to solve the synthetic aperture radar active target recognition problem by enabling agents to learn from historical image sequences. Compared to existing active target recognition methods, the DRL policy yields a significant improvement in recognition rates even under conditions of very scarce training samples. Furthermore, a DRL-based anti-jamming strategy design method was formulated in [30] to combat mainlobe jamming. By maximizing the probability of detection, the radar can effectively evade jamming attempts, regardless of whether the jammer employs individual or multiple jamming strategies. In contrast to [30], another study [31] focuses on developing intelligent jamming methods using DRL. Through the joint optimization of jamming type selection and power control by DRL, the solution not only demonstrates resilience in dynamic radar countermeasure scenarios but also achieves effective jamming performance. As a powerful optimization tool, DRL presents novel perspectives and methodologies for addressing complex optimization problems in RSP, offering vast potential for RSP applications. As evidenced by the literature above, the DRL-based RSP can enhance traditional RSP solutions and overcome their limitations. Motivated by the success of DRL in handling complex optimization problems in RSP, we believe that the AOA localization estimator design can benefit from the DRL strategy, since DRL provides a means for addressing ML optimization problem. To date, the DRL-based AOA target localization remains underexplored. Therefore, we intend to tackle the problem of MLE design for AOA localization from a DRL perspective for the first time. The main contributions are summarized as follows.

1) Learning-based framework to address the ML estimation problem: This paper introduces a novel framework to tackle the ML estimation problem. The framework offers an end-to-end optimization solution by directly learning optimal increments from bearing measurements. This eliminates the need for linearizing the objective function, which is a common step in traditional ML approaches [18]. The end-to-end optimization solution offered by the framework has several advantages. First, it simplifies the ML estimation process by directly learning the optimal increments, overcoming potential convergence issues associated with linearization. Second, this approach allows for a more accurate and robust estimation, especially in highly nonlinear systems where traditional ML approaches may struggle.

2) DRL-based location estimator to acquire the global optimal solution: Building on the learning-based framework, a DRL-based MLE is designed. In contrast to the GN estimator [18], the learning-based counterpart adeptly navigates and mitigates the convergence issue caused by the limitations of first-order linearization. Furthermore, the proposed estimator can effectively reduce bias compared to closed-form estimators such as the PLE [16], the TLS [12], and the WIVE [11]. This reduction in bias enhances the accuracy and reliability of the estimation process, making the learning-based estimator a favorable choice in practical applications. Additionally, the proposed method demonstrates its flexibility by successfully solving both 2D and 3D AOA localization problems. This capability highlights the robustness and applicability of the learning-based estimator across different dimensions, making it a valuable tool in various localization scenarios. To the best of our knowledge, this is the first work that adopts DRL to solve the AOA localization issue.

The paper is organized as follows. Section II defines the AOA localization problem and describes the ML estimation

algorithm implemented using the GN method. Section III introduces a learning-based framework to tackle the ML estimation problem and presents a DRL-based maximum likelihood estimator based on this framework. The performance analysis of the proposed estimator is also conducted. Section IV provides comprehensive simulation studies to demonstrate the effectiveness and superiority of the proposed estimator. The conclusions are drawn in Section V.

II. Problem Formulation

A. AOA Localization

Consider the problem of localizing an unknown deterministic target position in 2D space using AOA measurements from N sensors, where $N \geq 2$. Each sensor measures the bearing to the target in a noisy environment, and the bearing measurements are given by:

$$\tilde{\theta}_k = \theta_k + n_k, k = 1, \ldots, N, \tag{1}$$

where $\tilde{\theta}_k$ denotes the bearing measurement from sensor k, θ_k represents the true bearing from sensor k, and n_k is the bearing noise. We assume that n_k follows a white Gaussian distribution with zero mean and variance $\sigma^2_{n_k}$.

The objective of AOA localization is to determine the true target position based on noisy bearing measurements $\tilde{\theta}_1, \tilde{\theta}_2 \ldots, \tilde{\theta}_N$. Fig. 1 illustrates a typical geometry for AOA localization. Let $\boldsymbol{p} = [x, y]^T$ represent the target coordinate vector and $\boldsymbol{o}_k = [x_k, y_k]^T$ denote the position of sensor k, where the notation $[\cdot]^T$ signifies the transpose operator. The following nonlinear equation describes the relationship among the bearing angle, sensor position, and target position:

$$\theta_k = \arctan \frac{\Delta y_k}{\Delta x_k}, \tag{2}$$

where $\theta_k \in [0, 2\pi)$ represents the true bearing angle, $\Delta y_k = y - y_k$, and $\Delta x_k = x - x_k$.

B. Maximum Likelihood Estimation

The likelihood function for the bearing measurements is determined by the joint probability density function conditioned on the target location [32]:

$$P(\tilde{\boldsymbol{\theta}}|\boldsymbol{p}) = \frac{1}{(2\pi)^{N/2}|\boldsymbol{K}|} \times \exp\left\{-\frac{1}{2}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}(\boldsymbol{p}))^T \boldsymbol{K}^{-1}(\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}(\boldsymbol{p}))\right\}, \tag{3}$$

where

$$\tilde{\boldsymbol{\theta}} = [\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_N]^T$$

is the $N \times 1$ vector of noisy bearing measurements,

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1(\boldsymbol{p}) \\ \theta_2(\boldsymbol{p}) \\ \vdots \\ \theta_N(\boldsymbol{p}) \end{bmatrix} = \begin{bmatrix} \angle(\boldsymbol{p} - \boldsymbol{o}_1) \\ \angle(\boldsymbol{p} - \boldsymbol{o}_2) \\ \vdots \\ \angle(\boldsymbol{p} - \boldsymbol{o}_N) \end{bmatrix}$$

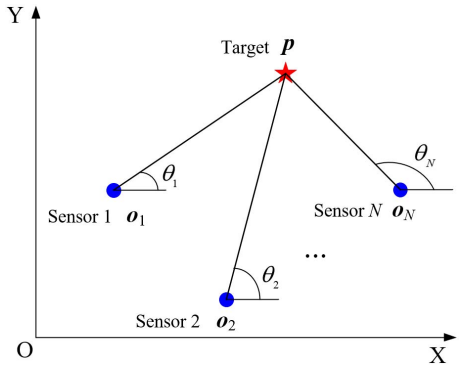


Fig. 1. 2D AOA localization geometry.

is the $N \times 1$ vector of bearing angles as a function of $\boldsymbol{p} = [x, y]^T$ with $\angle$ denoting the bearing angle, i.e., for $\boldsymbol{r} = [r_1, r_2]^T$,

$$\angle \boldsymbol{r} = \arctan \frac{r_2}{r_1},$$

$\boldsymbol{K} = diag\{\sigma^2_{n_1}, \ldots, \sigma^2_{n_N}\}$ is the $N \times N$ covariance matrix of the bearing measurement noise, and $|\boldsymbol{K}|$ denotes the determinant of $\boldsymbol{K}$.

Based on the principle of maximum likelihood estimation, the maximum likelihood estimate of the target position, denoted as $\hat{\boldsymbol{p}}_{\mathrm{MLE}}$, is obtained by maximizing the log-likelihood function $\ln P(\tilde{\boldsymbol{\theta}}|\boldsymbol{p})$ with respect to all possible target positions. Alternatively, it can be expressed as

$$\hat{\boldsymbol{p}}_{\mathrm{MLE}} = \arg\min_{\boldsymbol{p} \in \mathbb{R}^2} \Upsilon_{\mathrm{ML}}(\boldsymbol{p}), \tag{4}$$

where $\Upsilon_{\mathrm{ML}}(\boldsymbol{p})$ represents the ML cost function, given by

$$\Upsilon_{\mathrm{ML}}(\boldsymbol{p}) = \frac{1}{2}\boldsymbol{e}^T(\boldsymbol{p})\boldsymbol{K}^{-1}\boldsymbol{e}(\boldsymbol{p}), \boldsymbol{e}(\boldsymbol{p}) = \tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}(\boldsymbol{p}). \tag{5}$$

The minimization of $\Upsilon_{\mathrm{ML}}(\boldsymbol{p})$ over $\boldsymbol{p}$ is a challenging nonlinear least squares problem that lacks a closed-form solution. However, it is possible to obtain a numerical solution by utilizing iterative search algorithms such as the steepest descent algorithm, the Gauss-Newton algorithm, or the Nelder-Mead simplex algorithm. The Gauss-Newton (GN) algorithm, in particular, is commonly employed for computing the ML estimate. The GN algorithm uses the first-order Taylor series expansion to locally approximate the nonlinear objective function (5) with a linear one. The linearization step involves approximating $\boldsymbol{e}(\boldsymbol{p})$ with a linear function around the current estimate $\hat{\boldsymbol{p}}_i$:

$$\boldsymbol{e}(\boldsymbol{p}) \approx \boldsymbol{e}(\hat{\boldsymbol{p}}_i) + \boldsymbol{J}_i(\boldsymbol{p} - \hat{\boldsymbol{p}}_i), \tag{6}$$

where $\boldsymbol{J}_i$ is the Jacobian matrix of partial derivatives of $\boldsymbol{e}(\boldsymbol{p})$ with respect to $\boldsymbol{p}$, evaluated at $\hat{\boldsymbol{p}}_i$. Substituting this linearization into the objective function (5), we get the linearized objective function:

$$\bar{\Upsilon} = (\boldsymbol{e}(\hat{\boldsymbol{p}}_i) + \boldsymbol{J}_i(\boldsymbol{p} - \hat{\boldsymbol{p}}_i))^T \boldsymbol{K}^{-1}(\boldsymbol{e}(\hat{\boldsymbol{p}}_i) + \boldsymbol{J}_i(\boldsymbol{p} - \hat{\boldsymbol{p}}_i)). \tag{7}$$

By minimizing (7), we can obtain

$$\boldsymbol{J}_i^T \boldsymbol{K}^{-1} \boldsymbol{J}_i \boldsymbol{\zeta}_i = \boldsymbol{J}^T \boldsymbol{K}^{-1} \boldsymbol{e}_i, \tag{8}$$

where $\boldsymbol{\zeta}_i = [\zeta_{i,x}, \zeta_{i,y}]^T$ is the iteration increment, and

$$\boldsymbol{e}_i = [\tilde{\theta}_1 - \theta(\hat{\boldsymbol{p}}_i), \tilde{\theta}_2 - \theta(\hat{\boldsymbol{p}}_i), \ldots, \tilde{\theta}_N - \theta(\hat{\boldsymbol{p}}_i)]^T$$

is the residual vector at $\hat{\boldsymbol{p}}_i$. By matrix operation, $\boldsymbol{\zeta}_i$ can be obtained as

$$\boldsymbol{\zeta}_i = -[\boldsymbol{J}_i{}^T \boldsymbol{K}^{-1} \boldsymbol{J}_i]^{-1} \boldsymbol{J}_i{}^T \boldsymbol{K}^{-1} \boldsymbol{e}_i. \tag{9}$$

Therefore, the iteration step in the GN algorithm can be expressed as:

$$\hat{\boldsymbol{p}}_{i+1} = \hat{\boldsymbol{p}}_i + \boldsymbol{\zeta}_i. \tag{10}$$

The iteration of (10) continues until $||\boldsymbol{\zeta}_i||_2 < \epsilon$ or $i > S$ is met, where ϵ represents the limit of the iterative termination criteria, and S is the total number of iterations, which is associated with the required precision.

Problem *(Learning-based MLE for AOA localization):* The GN algorithm relies on local linear approximations of the nonlinear least squares objective function. However, these approximations may lead to ill-conditioning of the Jacobian matrix during the iteration process. Due to the potential ill-conditioning, the convergence of the GN algorithm becomes sensitive to the choice of the initial guess for the parameter vector $\boldsymbol{p}$. Such convergence difficulties can be aggravated in highly nonlinear systems because the local linearity assumption becomes less accurate. With the aforementioned considerations, our objective is to develop a learning-based maximum likelihood estimator. This estimator is designed to alleviate the linearization constraints inherent in traditional maximum likelihood estimation algorithms and determine the optimal increments $\boldsymbol{\zeta}_1^*, \boldsymbol{\zeta}_2^*, \ldots \boldsymbol{\zeta}_i^*$. Consequently, the task is reduced to the pursuit of the global optimum estimation for the parameter vector $\boldsymbol{p}$.

III. Learning-Based Estimator Design and Implementation

A. Learning Framework Construction

The discrete-time dynamical system (10) describes the transition procedure for an iterative estimator. Such transition procedure can be formulated as a Markov decision process (MDP) since the transition depends only on the current estimation $\hat{\boldsymbol{p}}_i$ and increment $\boldsymbol{\zeta}_i$. An MDP consists of a state space $\mathcal{S}$, action space $\mathcal{A}$, transition probability P, and a reward function R [33]. Particularly, these components are defined as follows:

1) State space $\mathcal{S}$: The state space is defined by

$$\mathcal{S} = \{s_i \triangleq (\hat{\boldsymbol{p}}_i, \tilde{\boldsymbol{\theta}}), i = 1, \ldots \bar{T}\}, \tag{11}$$

where s_i represents the state at time step i, and $\bar{T}$ denotes the time horizon.

2) Action space $\mathcal{A}$: The action space corresponds to the possible increments applied to the parameter estimates:

$$\mathcal{A} = \{a_i \triangleq \boldsymbol{\zeta}_i, i = 1, \ldots \bar{T}\}, \tag{12}$$

where a_i denotes the action at time step i.

3) Transition probability P: The transition probability refers to the probability of transitioning from state s_i to state s_{i+1} given a specific action a_i, denoted as

$$P(s_{i+1}|s_i, a_i). \tag{13}$$

4) Reward function R: The reward function assigns a numerical value r_i to each state-action pair (s_i, a_i), indicating the immediate reward associated with taking an action a_i in a particular state s_i. Here, the immediate reward is defined as

$$r_i \triangleq -\frac{1}{2} \boldsymbol{e}^T(\hat{\boldsymbol{p}}_i) \boldsymbol{K}^{-1} \boldsymbol{e}(\hat{\boldsymbol{p}}_i). \tag{14}$$

With above definitions, the agent's objective in the MDP is to learn a policy π that maximizes the expected cumulative rewards:

$$\mathbb{E}[r_1 + r_2 + \ldots + r_{\bar{T}} | a_1, a_2, \ldots, a_{\bar{T}}].$$

As such, the AOA localization problem can be reformulated as the following optimization problem:

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\bar{T}} \gamma^t r_t | s_t, a_t\right], \tag{15}$$

where $\gamma \in [0, 1]$ denotes the discount factor that determines the importance of future rewards relative to immediate rewards.

Reinforcement learning (RL) is an efficient approach for solving MDP problems [34], [35]. However, traditional RL algorithms often face the exploration-exploitation dilemma. Maximum entropy is regarded as a means to effectively augment the stability of exploration and exploitation in RL, with the objective of maximizing both the cumulative discounted reward and the expected entropy of the policy simultaneously [36]. By adopting the maximum entropy objective, the stochastic nature of the agent's policy can be significantly enhanced, thereby facilitating the exploration of a wider range of optimal decisions. Consequently, the reformulated AOA localization problem in (15) can be restated as follows:

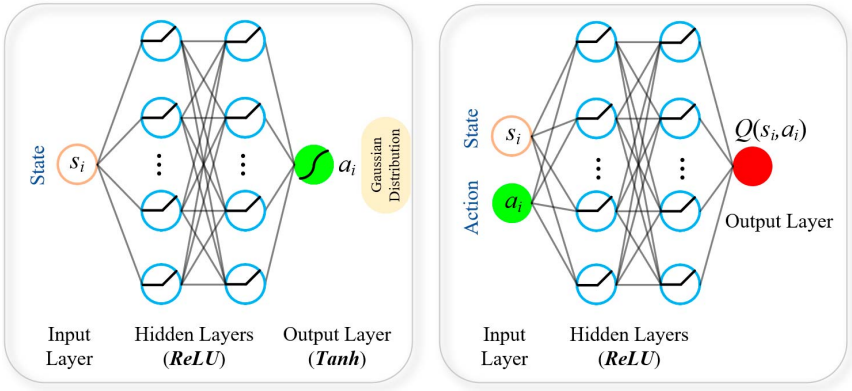
$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\bar{T}} \gamma^t (r_t + \alpha \mathcal{H}(\pi(\cdot|s_t)) | s_t, a_t\right], \tag{16}$$

where $\mathcal{H}(\cdot)$ represents the entropy that quantifies the uncertainty inherent in the policy, and α denotes the temperature parameter that governs the relative significance of both terms.

Soft policy iteration can be used to solve (16), which operates within the maximum entropy framework and alternates between soft policy evaluation and soft policy improvement [37], [38]. In the evaluation step, the algorithm seeks to compute the soft value of a fixed policy by optimizing the maximum entropy objective. Mathematically, this involves solving the modified Bellman equation:

$$\mathcal{T}^{\pi} Q(s_t, a_t) \triangleq r_t + \gamma \mathbb{E}_{s_{t+1} \sim P(s_{t+1}|s_t)}[V(s_{t+1})], \tag{17}$$

where $\mathcal{T}^{\pi}$ represents the soft Bellman backup operator, $Q(s_t, a_t)$ denotes the soft Q value of the state action pair



(a) Actor network (b) Critic network

Fig. 2. Network structures of actor and critic.

(s_t, a_t), and $V(s_t)$ signifies the soft state value of state s_t, defined as

$$V(s_t) = \mathbb{E}_{a_t \sim \pi(\cdot|s_t)}[Q(s_t, a_t) - \alpha \log \pi(a_t|s_t)]. \tag{18}$$

In the policy improvement step, the policy undergoes an update according to the following equation:

$$\pi_{new} = \arg\min_{\pi' \in \Pi} \mathscr{D}_{KL}\left(\pi'(\cdot|s_t), \frac{\exp\left(\frac{1}{\alpha} Q^{\pi_{old}}(s_t, a_t)\right)}{Z^{\pi_{old}}(s_t)}\right), \tag{19}$$

where Π represents a Gaussian policy set, π_{old} signifies the policy from the last update, $Q^{\pi_{old}}$ denotes the soft Q-value of π_{old}, and $Z^{\pi_{old}}$ serves as a normalization factor.

B. DRL-Based MLE for AOA Localization

Due to the continuous state and action spaces, we approximate the soft Q function $Q(s_t, a_t)$ and the policy $\pi(a_t|s_t)$ using deep neural networks (DNNs). The network structures are shown in Fig. 2, with the policy and the soft Q-function being parameterized by $\pi_{\boldsymbol{\psi}}(a_t|s_t)$ and $Q_{\boldsymbol{\omega}}(s_t, a_t)$, respectively. Notably, $\boldsymbol{\omega}$ and $\boldsymbol{\psi}$ represent the parameters of these networks. Utilizing the parameterized soft Q function and policy, the soft policy iteration in (17)-(19) transforms into the soft actor-critic (SAC) algorithm [37], [38], where the parameterized policy functions as the actor, while the parameterized soft Q-function serves as the critic. In the learning setup, the output of the actor network is modeled as a Gaussian distribution to ensure the tractability of the policy:

$$\pi_{\boldsymbol{\psi}} = \mathcal{N}(a_{\boldsymbol{\psi}}(s_t), \sigma^2_{\boldsymbol{\psi}}(s_t)), \tag{20}$$

where $a_{\boldsymbol{\psi}}(s_t)$ represents the estimation increment and $\sigma^2_{\psi}(s_t)$ denotes the standard deviation of the exploration noise.

The learning-based ML localization estimator, as depicted in Fig. 3, consists of an actor network denoted as $\pi_{\boldsymbol{\psi}}$, a pair of evaluation critic networks represented by $Q_{\boldsymbol{\omega}_m}$, and a pair of target critic networks denoted as $Q_{\bar{\boldsymbol{\omega}}_m}$, where $\forall m \in \{1, 2\}$. The actor network is responsible for decision-making, while the critic networks calculate a pair of Q-values to evaluate the policy. Note that the target critic networks are employed to alleviate positive bias in policy improvement and share identical structures with the evaluation critics. The learning procedure of the estimator is outlined in Algorithm 1. The main process can be further explained as follows.

1) Initialization: At the outset, the parameters of the actor and critic networks are initialized, and these parameters will be updated during the learning. Furthermore, an experience replay pool $\mathcal{M}$ is created.

2) Experience collection: The agent's experience is represented by a multidimensional tuple consisting of the chosen action, state transition, and feedback reward, denoted as (s_t, a_t, r_t, s_{t+1}). This tuple is acquired through the following interaction between the agent and environment. Initially, the agent randomly selects an initial state s_1 from the environment. Subsequently, the policy network $\pi_{\boldsymbol{\psi}}(\cdot|s_1)$ samples a random action a_1 based on the state s_1 and applies it to the environment to update (10). Then, the environment provides a reward signal r_1 based on (14) and transitions to the next state s_2. Finally, the formulated transition tuple (s_1, a_1, r_1, s_2) is stored in a replay pool $\mathcal{M}$. This process is iterated M times until $M = N_p$, where M denotes the total exploration step, and N_p represents the capacity of the replay pool. The formulated transition tuples are utilized for subsequent algorithmic parameter updates.

3) Parameter updating: After exploring the environment, the network parameters $\boldsymbol{\psi}$, $\boldsymbol{\omega}_m$ and $\bar{\boldsymbol{\omega}}_m$, $\forall m \in \{1, 2\}$ require optimization.

Firstly, the Bellman residual is defined as

$$e_{Q,m} = Q_{\boldsymbol{\omega}_m}(s_t, a_t) - y_t, \tag{21}$$

where

$$y_t = r_t + \gamma(\min_{m=1,2} Q_{\bar{\boldsymbol{\omega}}_m}(s_{t+1}, a_{t+1}) - \alpha \ln(\pi_{\boldsymbol{\psi}}(a_{t+1}|s_{t+1})))$$

denotes the target Q-value. Here, $\ln(\cdot)$ represents the function that yields the entropy value of the actions, $Q_{\boldsymbol{\omega}_m}$ and $Q_{\bar{\boldsymbol{\omega}}_m}$ correspond to the state-action Q-values computed by the evaluation and target critic networks, respectively. It is important to note that only the minimum of these two Q-values is utilized for the loss calculation to mitigate the biased Q value problem [39].

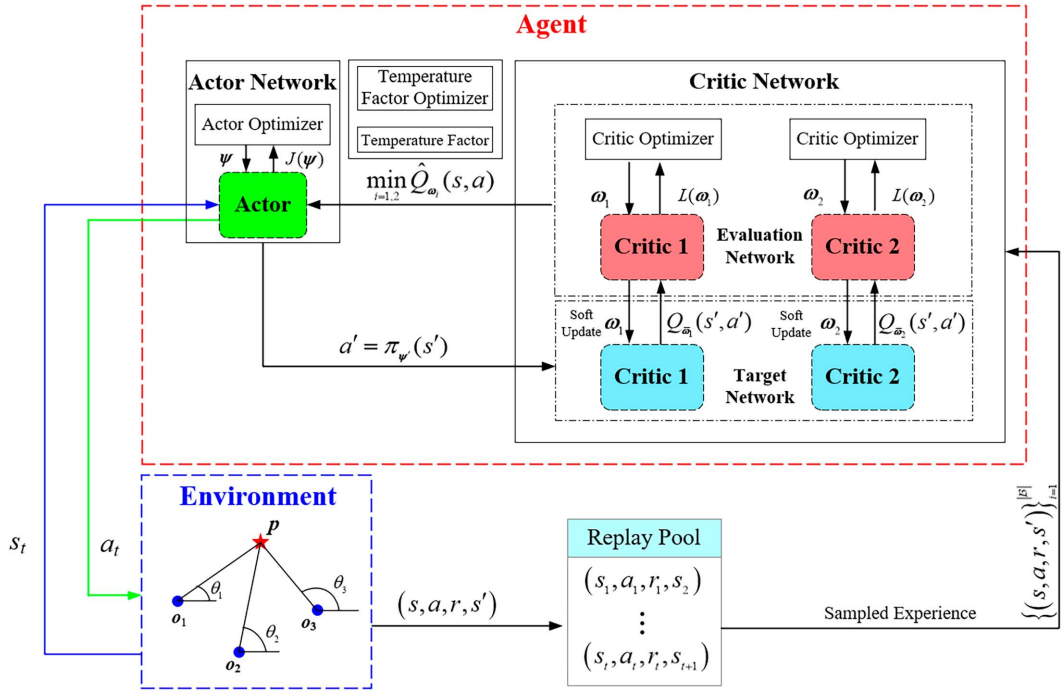


Fig. 3. Architecture of the DRL-based localization estimator.

Algorithm 1: Training process of the proposed learning-based MLE for AOA localization

Input: The message in (1), initial actor parameters ψ, critic parameters $\boldsymbol{\omega}_1$, $\boldsymbol{\omega}_2$, learning rates ι_Q, ι_π, ι_α, and empty replay pool $\mathcal{M}$

Output: Optimized netwrok parameters ψ^*, $\boldsymbol{\omega}_1^*$, $\boldsymbol{\omega}_2^*$

for *environment exploration* **do**
- Observe state s_t and select action $a_t \sim \pi_{\boldsymbol{\psi}}(\cdot|s_t)$
- Execute a_t and obtain r_t based on (14)
- Observe next state s_{t+1}
- Store (s_t, a_t, r_t, s_{t+1}) into $\mathcal{M}$

for *parameter updating* **do**
- Sample a batch of transitions $\mathcal{B}$ from $\mathcal{M}$
- Update critics by
- $\boldsymbol{\omega}_m \leftarrow \boldsymbol{\omega}_m - \iota_Q \nabla J_Q(\boldsymbol{\omega}_m)$, for $m = 1, 2$
- Update actor by
- $\boldsymbol{\phi} \leftarrow \boldsymbol{\phi} - \iota_\pi \nabla J_\pi(\boldsymbol{\phi})$
- Update temperature parameter by
- $\alpha \leftarrow \alpha - \iota_\alpha \nabla J_\alpha(\alpha)$
- Update target critics by
- $\bar{\boldsymbol{\omega}}_m \leftarrow \tau \bar{\boldsymbol{\omega}}_m + (1-\tau)\bar{\boldsymbol{\omega}}_m$, for $m = 1, 2$

Return Optimal parameters ψ^* and $\boldsymbol{\omega}_m^*$ for $m = 1, 2$

By minimizing the Bellman residual, the loss function for training the state-action value function can be formulated as:

$$L(\boldsymbol{\omega}_m) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{M}}\left[\frac{1}{2}e_{Q,m}^2\right], \tag{22}$$

where $(s_t, a_t) \sim \mathcal{M}$ represents a random sample of (s_t, a_t) from the replay pool $\mathcal{M}$. Consequently, the stochastic gradient can be obtained to update the parameters of the critic networks:

$$\begin{aligned}\nabla_{\boldsymbol{\omega}_m} L(\boldsymbol{\omega}_m) = \nabla_{\boldsymbol{\omega}_m} Q_{\boldsymbol{\omega}_m}(s_t, a_t) \cdot (Q_{\omega_m}(s_t, a_t) \\ - (r_t + \gamma(Q_{\bar{\boldsymbol{\omega}}_m}(s_{t+1}, a_{t+1}) \\ - \alpha \ln(\pi_{\boldsymbol{\psi}}(a_{t+1}|s_{t+1}))))).\end{aligned} \tag{23}$$

Next, the actor network can be updated by directly minimizing the expected KL-divergence in (19), and the objective function is defined as follows:

$$\begin{aligned}J(\boldsymbol{\psi}) = -\mathbb{E}_{s_t\sim\mathcal{M}}[\mathbb{E}_{a_t\sim\pi_{\boldsymbol{\phi}}}[\min_{m=1,2} Q_{\bar{\boldsymbol{\omega}}_m}(s_t, a_t) \\ - \alpha \log(\pi_{\boldsymbol{\psi}}(a_t|s_t))]].\end{aligned} \tag{24}$$

To address the challenge of computing the expectation over the policy's output distribution, the policy is reparameterized through a neural network transformation denoted as $g_{\boldsymbol{\psi}}(\epsilon_t, s_t)$. This allows rewriting (24) as:

$$\begin{aligned}J(\boldsymbol{\psi}) = -\mathbb{E}_{s_t\sim\mathcal{M},\boldsymbol{\epsilon}_t\sim\mathcal{N}}[\min_{m=1,2} Q_{\bar{\boldsymbol{\omega}}_m}(s_t, g_{\boldsymbol{\psi}}(\epsilon_t, a_t)) \\ -\alpha \ln(\pi_{\boldsymbol{\psi}}(g_{\boldsymbol{\psi}}(\epsilon_t, s_t)|s_t))],\end{aligned} \tag{25}$$

where ϵ_t is sampled from a Gaussian distribution. Thus, the gradient of the policy can be calculated as:

$$\begin{aligned}\nabla_{\boldsymbol{\psi}} J(\boldsymbol{\psi}) = \nabla_{\boldsymbol{\psi}} \alpha \ln(\pi_{\psi}(g_{\boldsymbol{\psi}}(\epsilon_t, s_t)|s_t)) \\ + (\nabla_{a_t} \alpha \ln(\pi_{\boldsymbol{\psi}}(a_t|s_t)) \\ - \nabla_{a_t} \min_{m=1,2} Q_{\bar{\boldsymbol{\omega}}_m}(s_t, a_t))\nabla_{\boldsymbol{\psi}} g_{\boldsymbol{\psi}}(\epsilon_t, s_t).\end{aligned} \tag{26}$$

Moreover, instead of manually selecting the weight α, the temperature parameter is adaptively adjusted to enhance the learning performance. The loss function for the temperature parameter is given by:

$$J(\alpha) = \mathbb{E}_{a_t\sim\pi_{\boldsymbol{\psi}}}[-\alpha \ln \pi_{\boldsymbol{\psi}}(a_t|s_t) - \alpha\bar{\mathcal{H}}], \tag{27}$$

where $\bar{\mathcal{H}}$ is a constant vector representing the target entropy.

The gradient of the loss function with respect to the temperature parameter can be derived as follows:

$$\nabla_\alpha J(\alpha) = -\ln \pi(a_t|s_t) - \bar{\mathcal{H}}. \tag{28}$$

Finally, to ensure the stability of the learning process, the parameters of the target critic networks are updated from the evaluation critics' parameters using a soft-updating method:

$$\boldsymbol{\omega}_m = \tau \boldsymbol{\omega}_m + (1-\tau)\bar{\boldsymbol{\omega}}_m, \forall m \in \{1,2\}, \tag{29}$$

where $\tau \in (0,1)$ represents the update factor.

Upon completing the training, the optimal policy parameters $\boldsymbol{\phi}^*$ can be obtained from Algorithm 1. Consequently, the optimal estimation increment $\boldsymbol{\zeta}_i^*$ can be acquired. By substituting $\boldsymbol{\zeta}_i^*$ into (10), the position vector $\hat{\boldsymbol{p}}_{i+1}^*$ is obtained. The localization task ends when $||\boldsymbol{\zeta}_i^*||_2$ is smaller than $\epsilon_\#$ or $i > \bar{T}$, where $\epsilon_\#$ denotes the error threshold and $\bar{T}$ is the time horizon of the MDP.

C. Performance Analysis

1) Global optimum analysis: As per (14), the reward r_t is non-positive and is bounded due to noise. Consequently, there exists an interval defined as:

$$r_t \in [r_{\min}, 0], \tag{30}$$

where $r_{\min}$ represents the utmost lower limit for the reward function.

According to (30), we can derive the subsequent Lemma 1 and Lemma 2 to facilitate the analysis of the algorithm's convergence.

Lemma 1: Let $\mathcal{T}^\pi$ denote the soft Bellman backup operator as defined in (17). Additionally, let $Q^0 : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ be a mapping with $|\mathcal{A}| < \infty$. We define the sequence Q^{k+1} recursively as $Q^{k+1} = \mathcal{T}^\pi Q^k$, where k indexes the iterations. Then as $k \to \infty$, the sequence Q^k converges to the soft Q-value associated with policy π.

Proof: Define the entropy augmented reward function as follows:

$$\bar{r}_t \triangleq r_t - \gamma \mathbb{E}_{s_{t+1}\sim P}[\mathbb{E}_{a_{t+1}\sim\pi}[\alpha \ln(\pi(a_{t+1}|s_{t+1}))]], \tag{31}$$

then we have

$$\begin{aligned} &r(s_t,a_t) + \gamma \mathbb{E}_{s_{t+1}\sim P}[V(s_{t+1})] \\ &= r(s_t,a_t) + \gamma \mathbb{E}_{s_{t+1}}[\mathbb{E}_{a_{t+1}\sim\pi}[Q(s_{t+1},a_{t+1}) - \alpha \log \pi(a_{t+1}|s_{t+1})]] \\ &= r(s_t,a_t) + \gamma \mathbb{E}_{s_{t+1},a_{t+1}}[Q(s_{t+1},a_{t+1})] + \alpha\gamma \mathbb{E}_{s_{t+1}}[\mathcal{H}(\pi(\cdot|s_{t+1}))] \\ &= \bar{r}_t + \gamma \mathbb{E}_{s_{t+1}\sim P, a_{t+1}\sim\pi}[Q(s_{t+1},a_{t+1})]. \end{aligned} \tag{32}$$

Therefore, the soft Bellman backup update rule can be reformulated as:

$$\mathcal{T}^\pi Q_\pi(s_t,a_t) = \bar{r}_t + \gamma \mathbb{E}_{s_{t+1}\sim P, a_{t+1}\sim\pi}[Q_\pi(s_{t+1},a_{t+1})]. \tag{33}$$

Given that $|\mathcal{A}| = |a| < \infty$, the second term in (31) is bounded. In accordance with (33), there exist lower and upper bounds, denoted as $\bar{r}_{\min}$ and $\bar{r}_{\max}$ respectively, such that $\bar{r}_t \in [\bar{r}_{\min}, \bar{r}_{\max}]$. Additionally, $|\bar{r}_t| \le \tilde{r}$ with $\tilde{r} = \max\{|\bar{r}_{\min}|, |\bar{r}_{\max}|\}$. Moreover,

$$Q_\pi(s_t,a_t) = \bar{r}_t + \gamma \sum_{t+1}^{\infty} \sum_{a_{t+1}} \pi(a_{t+1}|s_{t+1}) \sum_{s_{t+1}} P(s_{t+1}|s_t)\bar{r}_{t+1}. \tag{34}$$

Therefore, we can deduce that

$$||Q_\pi(s_t,a_t)||_\infty \le \bar{r}/(1-\gamma), \tag{35}$$

where $||Q_\pi(s,a)||_\infty = \max_{s,a}|Q_\pi(s,a)|$. Consequently, the Q-value Q_π is bounded in ∞-norm. For any two vectors Q and Q', the following inequality holds:

$$\begin{aligned} &||\mathcal{T}^\pi Q - \mathcal{T}^\pi Q'|| \\ &= ||r_\pi(s_t,a_t) + \gamma \mathbb{E}_{s_{t+1}\sim P(s_{t+1}|s_t,a_t)}[Q(s_{t+1},\cdot)] \\ &\quad - r_\pi(s_t,a_t) - \gamma \mathbb{E}_{s_{t+1}\sim P(s_{t+1}|s_t,a_t)}[Q'(s_{t+1},\cdot)]||_\infty \\ &= ||\gamma \mathbb{E}_{s_{t+1}\sim P(s_{t+1}|s_t,a_t)}[Q(s_{t+1},\cdot) - Q'(s_{t+1},\cdot)]||_\infty \\ &= ||\sum_{s_{t+1}} \gamma(Q(s_{t+1},\cdot) - Q'(s_{t+1},\cdot))P(s_{t+1}|s_t,a_t)||_\infty \\ &\le \sum_{s_{t+1}} \gamma ||Q(s_{t+1},\cdot) - Q'(s_{t+1},\cdot)||_\infty P(s_{t+1}|s_t,a_t) \\ &= \gamma ||Q - Q'||_\infty, \end{aligned} \tag{36}$$

where Q and Q' are the Q values approximated at the last and current iterations, respectively. According to Banach fixed-point theorem, the soft Bellman backup operator is a γ-contraction with $0 \le \gamma \le 1$. Therefore, iterative soft policy evaluation will converge on the unique fixed point of $\mathcal{T}^\pi$. Since $\mathcal{T}^\pi Q_\pi = Q_\pi$ is a fixed point, we can conclude that iterative policy evaluation converges on Q_π. Therefore, the sequence Q^k will converge to Q_π as $k \to \infty$.

Lemma 2: Given the the last updated policy $\pi_{old} \in \Pi$ and the new policy π_{new} derived from (19), it can be stated that $Q^{\pi_{new}}(s,a) \ge Q^{\pi_{old}}(s,a)$ holds for $\forall s \in \mathcal{S}$ and $\forall a \in \mathcal{A}$ with $|\mathcal{A}| < \infty$.

Proof: Let π_{old} denote an arbitrary policy within the policy space Π, with $Q^{\pi_{old}}$ and $V^{\pi_{old}}$ representing the corresponding soft state-action value function and soft state value function under π_{old}, respectively. Furthermore, let π_{new} be a policy defined as:

$$\begin{aligned} &\pi_{new}(\cdot|s_t) \\ &= \arg\min_{\pi'\in\Pi} \mathscr{D}_{KL}(\pi'(\cdot|s_t)||\exp(Q^{\pi_{old}}(s_t,\cdot) - \log Z^{\pi_{old}}(s_t)) \\ &= \arg\min_{\pi'\in\Pi} J_{\pi_{old}}(\pi'(\cdot|s_t)). \end{aligned} \tag{37}$$

Since we can always choose $\pi_{new} = \pi_{old} \in \Pi$, the inequality $J_{\pi_{old}}(\pi_{new}(\cdot|s_t)) \le J_{\pi_{old}}(\pi_{old}(\cdot|s_t))$ must be satisfied. Given this, we have

$$\begin{aligned} &\mathbb{E}_{a_t\sim\pi_{new}}[\log \pi_{new}(a_t|s_t) - Q^{\pi_{old}}(s_t,a_t) + \log Z^{\pi_{old}}(s_t)] \\ &\le \mathbb{E}_{a_t\sim\pi_{old}}[\log \pi_{old}(a_t|s_t) - Q^{\pi_{old}}(s_t,a_t) + \log Z^{\pi_{old}}(s_t)]. \end{aligned} \tag{38}$$

In view of the partition function $Z^{\pi_{old}}$ being dependent solely on the state, the aforementioned inequality can be simplified to

$$\mathbb{E}_{a_t\sim\pi_{new}}[Q^{\pi_{old}}(s_t,a_t) - \log \pi_{new}(a_t|s_t)] \ge V^{\pi_{old}}(s_t). \tag{39}$$

By virtue of the soft Bellman equation, it consequently follows that

$$\begin{aligned} &Q^{\pi_{old}}(s_t,a_t)\\ &\quad= r_t+\gamma\mathbb{E}_{s_{t+1}\sim P}[V^{\pi_{old}}(s_{t+1})]\\ &\quad\le r_t+\gamma\mathbb{E}_{s_{t+1}\sim P}[\mathbb{E}_{a_{t+1}\sim\pi_{new}}[Q^{\pi_{old}}(s_{t+1},a_{t+1})\\ &\qquad-\log\pi_{new}(a_{t+1}|s_{t+1})]]\\ &\quad\vdots\\ &\quad\le Q^{\pi_{new}}(s_t,a_t), \end{aligned} \tag{40}$$

where (39) is repeatedly employed and thus omitted. Hence, the proof is concluded. The convergence to $Q^{\pi_{new}}$ can be inferred from Lemma 1.

A theorem is derived based on Lemma 1 and Lemma 2 to demonstrate that the convergence of Algorithm 1.

Theorem 1: Let π_0 be any initial policy in Π, and let π_i represent the policy obtained at the i-th policy improvement step. Through iterative application of soft policy evaluation and soft policy improvement steps, π_i converges to the optimal policy π_* as $i\to\infty$ such that $Q^{\pi_*}(s_t,a_t)\ge Q^{\pi}(s_t,a_t)$ for all $\pi\in\Pi$ and $(s_t,a_t)\in\mathcal{S}\times\mathcal{A}$ with $|\mathcal{A}|<\infty$.

Proof: By Lemma 2, we know that the policy can achieve a better estimation performance after each policy improvement, that is, $Q^{\pi_{i-1}}\le Q^{\pi_i}$. Since Q^{π} is bounded above for $\pi\in\Pi$ due to the boundedness of both the reward and entropy, the sequence π_i converges to a specific policy π^*. Upon convergence, it is guaranteed that $J_{\pi^*}(\pi^*(\cdot|s_t))<J_{\pi^*}(\pi(\cdot|s_t))$ for all $\pi\in\Pi$, $\pi\ne\pi^*$. Consequently, we can establish the following inequality:

$$\mathbb{E}_{s_t\sim\pi}[Q^{\pi^*}(s_t,a_t)-\log\pi^*(a_t|s_t)]>V^{\pi}(s_t). \tag{41}$$

Applying the same iterative process as in the proof of Lemma 2, and referring to (41), we establish that $Q^{\pi^*}(s_t,a_t)>Q^{\pi}(s_t,a_t)$ holds for all $s_t\in\mathcal{S}$ and $a_t\in\mathcal{A}$. This implies that the soft value associated with any alternative policy in Π is lower than that of the converged policy. Consequently, we conclude that π^* is the optimal policy within the set Π.

2) Cramér-Rao lower bound (CRLB): The CRLB serves as a benchmark for the best achievable accuracy in parameter estimation [40]. Mathematically, for a parameter $\boldsymbol{p}$, the CRLB is bounded by the inverse of the Fisher information:

$$\mathcal{C}_{\boldsymbol{p}}=(\mathcal{J}^T\boldsymbol{K}^{-1}\mathcal{J})^{-1}, \tag{42}$$

where $\mathcal{J}$ is the $N\times 2$ Jacobian matrix of $\tilde{\boldsymbol{\theta}}-\boldsymbol{\theta}(\boldsymbol{p})$ with respect to $\boldsymbol{p}$,

$$\mathcal{J}=-\begin{bmatrix}\frac{\partial\theta_1}{\partial\boldsymbol{p}^T}\\ \vdots\\ \frac{\partial\theta_N}{\partial\boldsymbol{p}^T}\end{bmatrix}=\begin{bmatrix}-\frac{\boldsymbol{u}_1^T}{d_1}\\ \vdots\\ -\frac{\boldsymbol{u}_N^T}{d_N}\end{bmatrix}, \tag{43}$$

with

$$\boldsymbol{u}_n=[\sin\theta_n,-\cos\theta_n]^T, \tag{44}$$

$$d_n=\sqrt{(x-x_n)^2+(y-y_n)^2}. \tag{45}$$

Therefore, the location error of $\boldsymbol{p}$ satisfies the relationship of

$$\mathbb{E}\left\{\|\hat{\boldsymbol{p}}-\boldsymbol{p}\|^2\right\}\ge[\mathcal{C}_{\boldsymbol{p}}]_{1,1}+[\mathcal{C}_{\boldsymbol{p}}]_{2,2}. \tag{46}$$

3) Complexity analysis: The complexity analysis of Algorithm 1 is as follows. According to [41] and [42], the complexity of the SAC algorithm mainly depends on the complexity of training the actor and critic networks. Let L_a and L_c represent the number of fully connected layers in the actor and critic networks, respectively. Additionally, let $u_a^{(i)}$ and $u_c^{(j)}$ denote the number of neurons in the i-th layer of actor and the j-th layer of critic, respectively. The computational complexity of the actor network is

$$O\left(\sum_{i=1}^{L_a}u_a^{(i)}\cdot u_a^{(i+1)}\right). \tag{47}$$

Similarly, the computational complexity of the critic network is

$$O\left(\sum_{j=1}^{L_c}u_c^{(j)}\cdot u_c^{(j+1)}\right). \tag{48}$$

Note that only the critic network needs to be trained, while the target network utilizes a soft update to copy the weights from the critic network. Therefore, the computational complexity of Algorithm 1 during the training period is

$$O\left(M\cdot\left(\sum_{i=1}^{L_a}u_a^{(i)}\cdot u_a^{(i+1)}+2\cdot\sum_{j=1}^{L_c}u_c^{(j)}\cdot u_c^{(j+1)}\right)\right), \tag{49}$$

where M stands for the number of episodes in the training process. It is evident that the computational complexity is linear with respect to the number of training episodes and the depth of the hidden layers. Moreover, as demonstrated in Section IV-B, the execution time of the DRL-based estimator is acceptable compared to other iterative estimators.

IV. Simulation Results

This section presents the performance analysis of the proposed learning-based MLE. First, to demonstrate the effectiveness of SAC algorithm, we compare it to other prominent DRL algorithms in solving the formulated MDP problem. Next, we conduct a comparative analysis of the SAC-based estimator versus conventional AOA localization estimators. Finally, we extend our SAC-based estimator to 3D and evaluate its performance against baseline 3D estimators. All experiments were conducted using a computer with a 3.00GHz Intel Core i5 processor, 16 GB RAM, and an Nvidia GeForce RTX 3060 GPU. For different estimators, the performance evaluation is carried out in terms of bias norm (BNorm) and root mean squared error (RMSE) [32]. The BNorm and RMSE are computed as

$$\text{BNorm}=\left\|\frac{1}{\Xi}\sum_{k=1}^{\Xi}(\hat{\boldsymbol{p}}-\boldsymbol{p})\right\|_2 \tag{50}$$

TABLE I
HYPER-PARAMETERS OF DIFFERENT DRL ALGORITHMS

Hyperparameters	DDPG	TD3	SAC	TRPO	PPO
Policy distribution	-	-	Normal	Beta	Beta
Actor learning rate	1e-4	1e-4	1e-4	-	1e-4
Critic learning rate	1e-4	1e-4	1e-4	1e-4	1e-4
Batch size	256	256	256	256	256
Discount factor	1	1	1	1	1
Actor hidden units	256	256	256	256	256
Critic hidden units	256	256	256	256	256
Exploration noise std	0.2	0.2	-	-	-
Policy noise std	-	0.2	-	-	-
Soft update factor	5e-3	5e-3	5e-3	-	-
GAE factor	-	-	-	0.95	0.95
Clip factor	-	-	-	-	0.2
Update steps	-	-	-	-	10
KL-divergence limit	-	-	-	5e-5	-
Backtracking steps	-	-	-	10	-
Backtracking coefficient	-	-	-	0.5	-

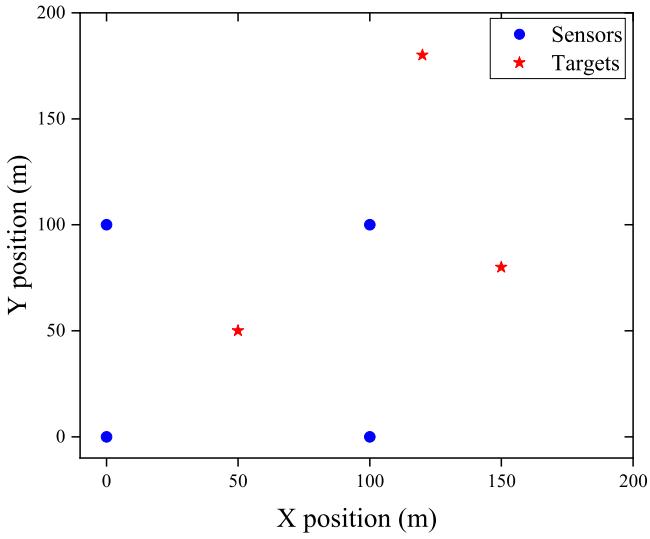


Fig. 4. 2D simulated AOA localization geometry.

and

$$\mathrm{RMSE} = \left(\frac{1}{\Xi} \sum_{k=1}^{\Xi} \|\hat{\boldsymbol{p}} - \boldsymbol{p}\|_2^2 \right)^{1/2}, \tag{51}$$

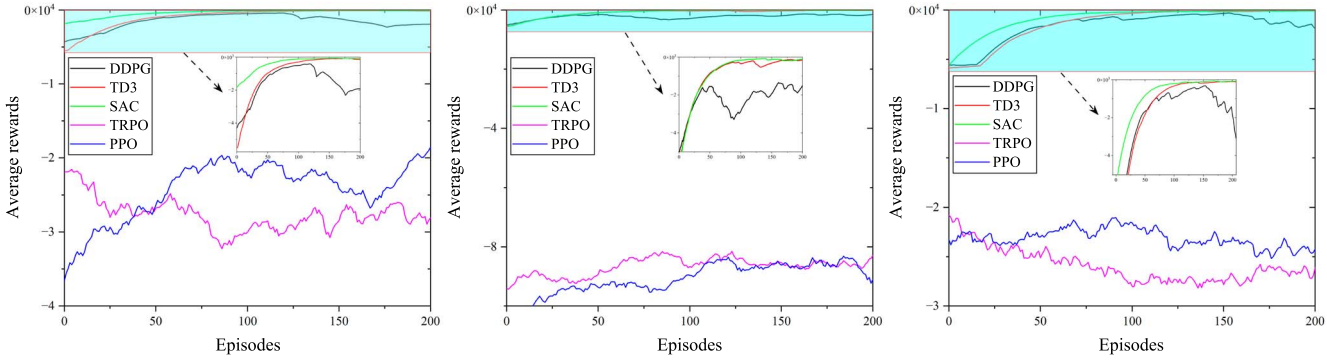
where $\hat{\boldsymbol{p}}$ is the estimate of $\boldsymbol{p}$ and $\Xi = 10000$ is the number of Monte Carlo runs.

A. Performance Analysis of Different DRL Algorithms

DRL methods can be classified as on-policy or off-policy methods, depending on whether the learning process directly updates the current policy. Notable off-policy techniques, such as DDPG [43], TD3 [44], and SAC optimize a target policy using experiences generated by a separate behavior policy. In contrast, well-known on-policy methods, like TRPO [45] and PPO [46], directly update the behavior policy during training. These algorithms have proven successful across diverse domains. To ensure the objectivity and comparability of the experimental results, we select DDPG, TD3, TRPO, and PPO as the benchmark algorithms for comparison with SAC. The hyper-parameters of these algorithms used in the experiments are reported in Table I.

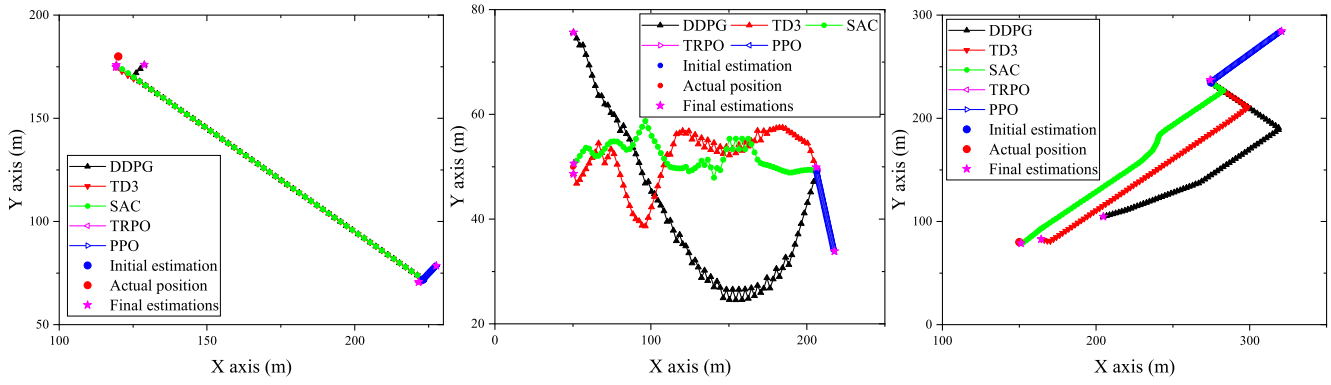
Fig. 4 illustrates the simulated 2D AOA localization geometry. Three targets located at coordinates (120m, 180m), (250m, 50m), and (50m, 50m), along with four sensors positioned at (0m, 0m), (0m, 100m), (100m, 100m), and (100m, 0m), are utilized to perform the localization tasks. Note that, in this study, we use fixed sensor-target geometries to evaluate the performance of various algorithms. It is well-known that the relative sensor-target geometry significantly impacts the efficacy of an estimator. For an in-depth optimality analysis of sensor-target localization geometries, please refer to [47], [48], [49]. The noise in bearing measurements is assumed to be independent and identically distributed (i.i.d.) with $\sigma_{n_1}^2 = \cdots = \sigma_{n_N}^2 = \sigma^2$. The learning curves of the five DRL algorithms at $\sigma = 0.5$ for the three targets are given in Fig. 5. Clearly, TRPO and PPO show oscillation and fail to achieve convergence in all localization tasks. This is because TRPO and PPO assume a stationary environment, which does not hold in dynamic localization tasks. The non-stationary nature of the tasks causes these on-policy methods to struggle with adapting quickly to changing conditions, leading to instability. Additionally, on-policy algorithms require new data for each update, making them sample inefficient and further exacerbating convergence issues [50]. In contrast, off-policy algorithms demonstrate better convergence. Among them, DDPG exhibits the most fluctuation in reward curves due to its severe overestimation bias. TD3 addresses this issue by reducing the overestimation bias inherent in DDPG, resulting in convergence curves similar to those of SAC. Additionally, as shown in Fig. 5, SAC achieves faster convergence and greater stability. This superior performance is attributed to SAC's entropy regularization, which encourages exploration and enhances robustness in non-stationary environments.

To provide a more detailed explanation, we present the DRL-based localization processes for the three targets. The estimated trajectories for the three targets, obtained using five DRL algorithms, are depicted in Fig. 6(a)–6(c), where the initial estimates are randomly selected. Evidently, the trajectories estimated by off-policy algorithms gradually converge towards the true target positions, while those estimated by on-policy algorithms do not exhibit a similar tendency towards the targets. This observation further emphasizes the effectiveness of off-policy algorithms and the limitations of on-policy algorithms in addressing the proposed MDP framework. Define the localization error of target $j \in \{1, 2, 3\}$ at iteration step k as $error_{j,k} = \sqrt{(\hat{x}_{j,k} - x_j)^2 + (\hat{y}_{j,k} - y_j)^2}$. The localization errors of target $j \in \{1, 2, 3\}$ are illustrated in Fig. 6(d)–6(f). Correspondly, the reward curves of target $j \in \{1, 2, 3\}$ are shown in Fig. 6(g)–6(i), respectively. From Fig. 6(d)–6(i), it is apparent that SAC accurately estimates the position information of the three targets, as

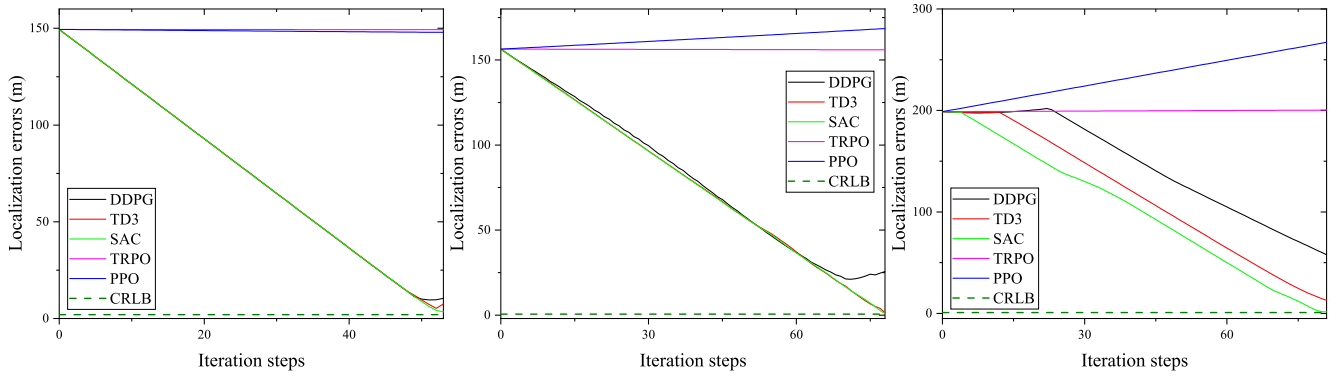


(a) Learning curves for target 1 (b) Learning curves for target 2 (c) Learning curves for target 3

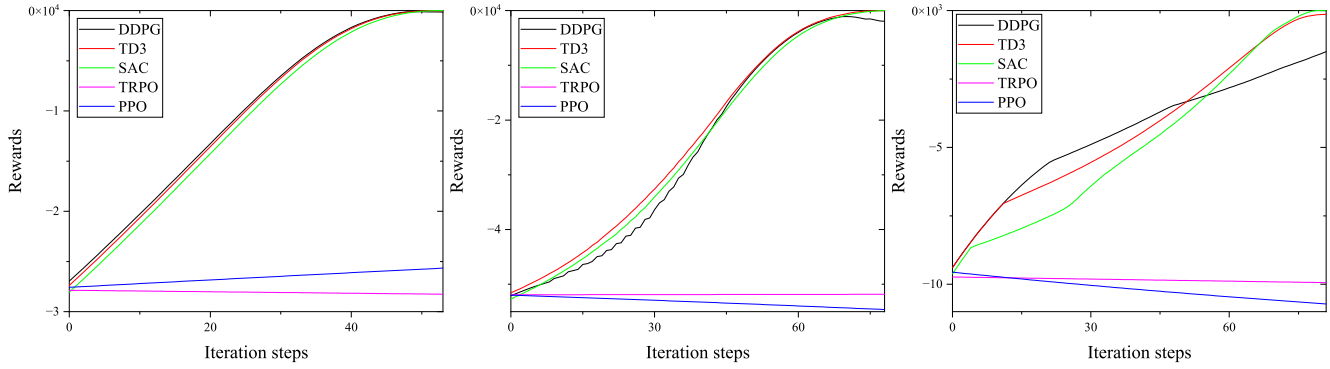
Fig. 5. Learning curves of different DRL algorithms.



(a) Localization trajectories for target 1 (b) Localization trajectories for target 2 (c) Localization trajectories for target 3

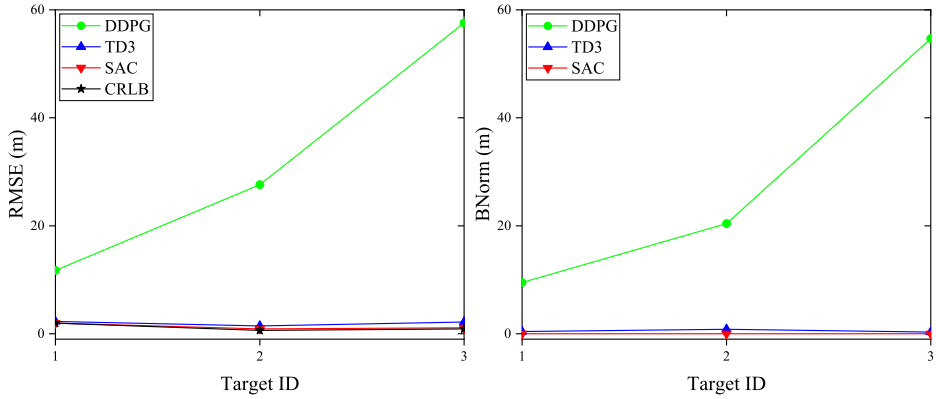


(d) Localization errors and CRLB for target 1 (e) Localization errors and CRLB for target 2 (f) Localization errors and CRLB for target 3



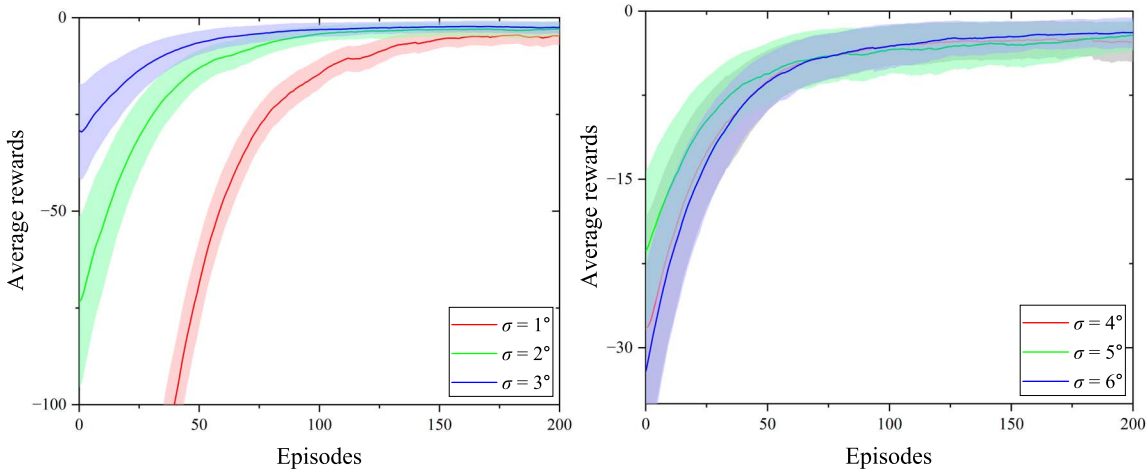
(g) Rewards for target 1 (h) Rewards for target 2 (i) Rewards for target 3

Fig. 6. Performance analysis of different DRL algorithms.



(a) RMSE versus target ID (b) BNorm versus target ID

Fig. 7. RMSE and BNorm of position estimate for different off-policy DRL-based estimators.



(a) Learning curves for $\sigma = 1^\circ \sim 3^\circ$ (b) Learning curves for $\sigma = 4^\circ \sim 6^\circ$

Fig. 8. Effectiveness of the proposed DRL-based MLE for 2D AOA localization.

the localization errors for each target closely match the CRLB, and the rewards reach their maximum value. Although both DDPG and TD3 algorithms achieve convergence, they exhibit oscillation and struggle to reach the optimal solution compared to SAC. This difference arises from the distinct exploration modes employed by SAC and DDPG/TD3. SAC incorporates policy entropy into the optimization function to encourage more thorough exploration of the action space, facilitating smoother convergence to the optimal solution. In contrast, DDPG and TD3 rely solely on adding noise to enhance exploration [51]. This approach may lead to less efficient exploration of the action space and, as a result, less efficient convergence to the optimal solution.

Fig. 7(a) and 7(b) illustrate the RMSE and BNorm metrics for the three targets using the DDPG, TD3, and SAC algorithms. Evidently, the SAC algorithm outperforms the others, exhibiting the lowest RMSE and BNorm values. The TD3 algorithm follows, performing slightly worse than SAC, while the DDPG algorithm shows the poorest performance among the three off-policy algorithms. The above results indicate that SAC performs best for addressing the proposed MDP framework, exhibiting greater stability and better convergence compared to other DRL algorithms. Therefore, this paper employs SAC to solve the AOA localization problem.

B. Comparison With Traditional AOA Localization Estimators

In this section, two numerical cases are presented to illustrate the performance of the proposed learning-based MLE. We initially assess the performance of PLE [16], TLS [12], WIVE [11], and GN [18]. Subsequently, we extend our estimator to 3D and compare it with OVE [13], 3D-PLE [13], 3D-WIVE [13], convergent iteration method (termed as CIM here) in [6], and GN [18]. The weighting vectors of WIVE and 3D-WIVE are initialized by PLE and 3D-PLE, respectively. The initial guess of GN is set to be the result of PLE (3D-PLE for 3D) to avoid divergence. The number of iterations for the GN algorithm is set to 10, while for the CIM algorithm, it is set to 1000.

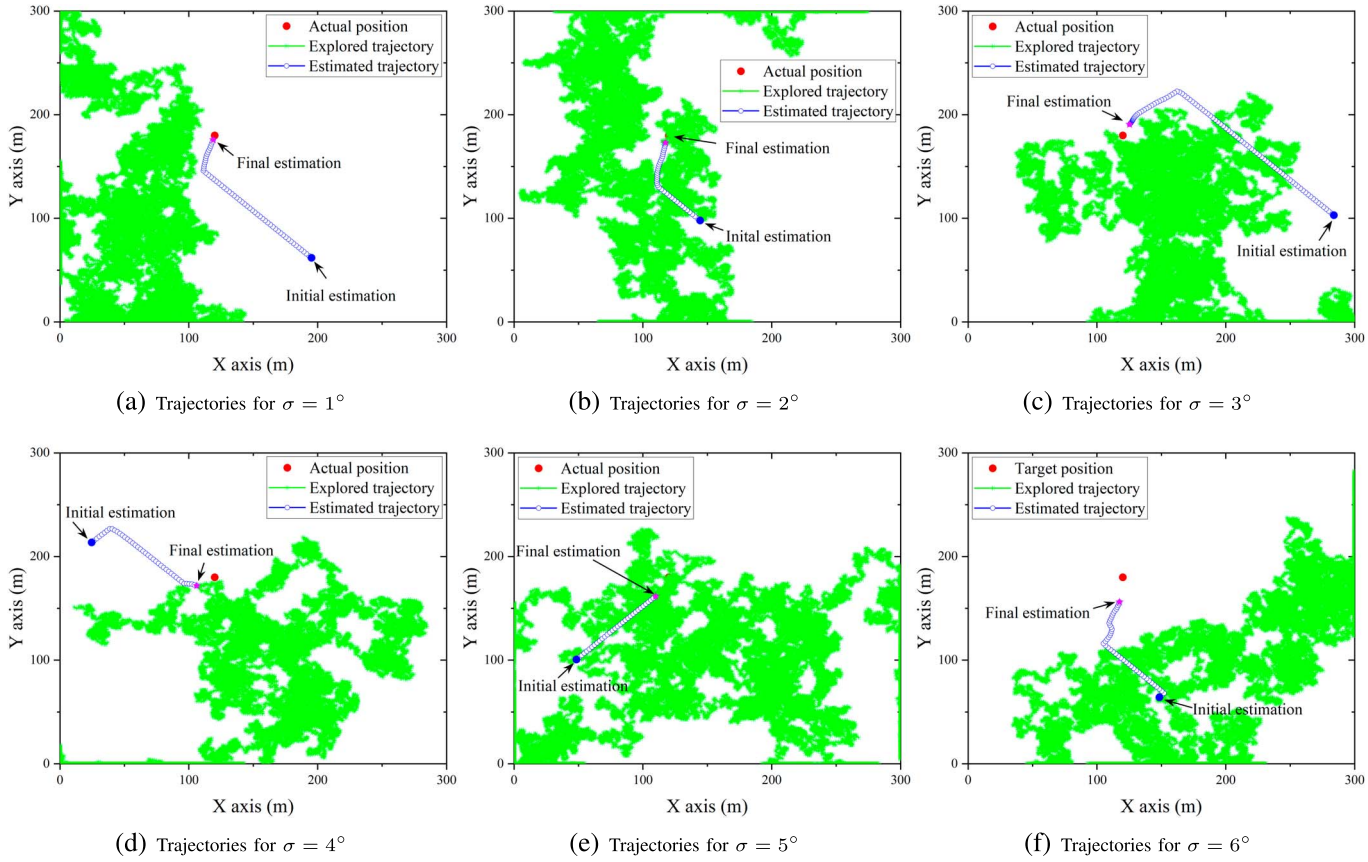
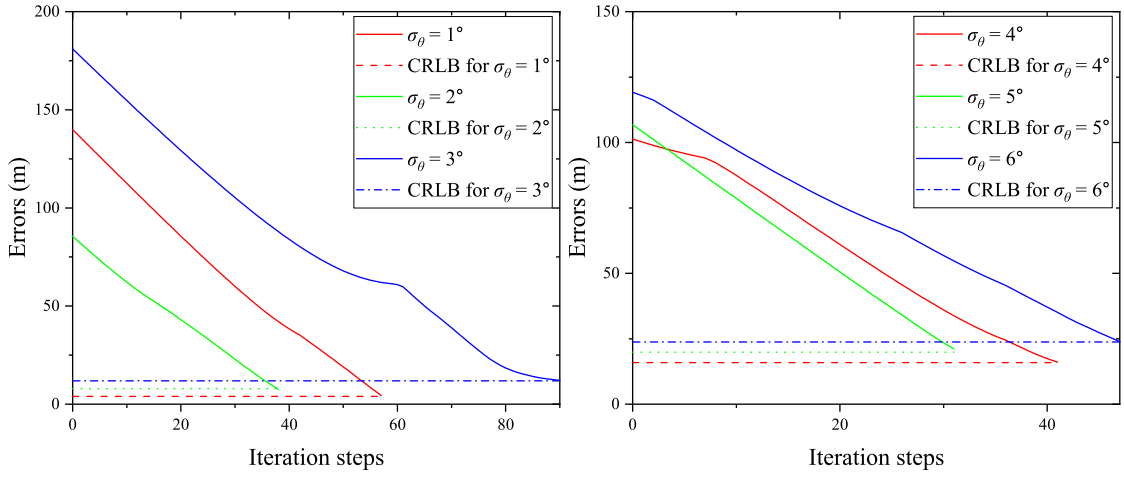


Fig. 9. Description of agent and environment interaction in 2D space.

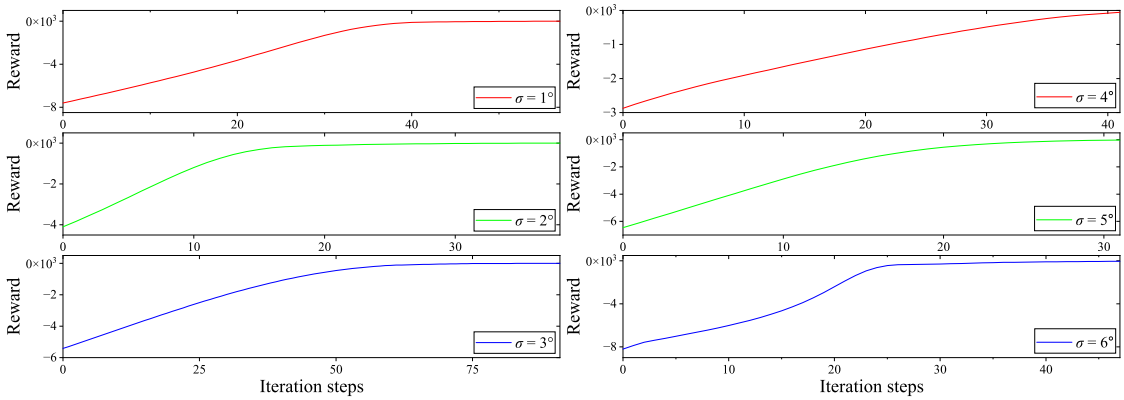
1) 2D AOA Localization: Here, we utilize the sensor network structure described in the previous section and select target 1 as an illustrative example. The learning curves of the proposed estimator for $\sigma \in \{1^\circ, 2^\circ, 3^\circ, 4^\circ, 5^\circ, 6^\circ\}$ are depicted in Fig. 8. It is observed that the proposed estimator achieves convergence across different levels of noise. Fig. 9(a)–9(f) present the trajectories resulting from algorithmic exploration, and the trajectories estimated using the proposed estimator. Clearly, the explored trajectories are stochastically selected, and the estimated trajectories tend to approach the actual position points even with increased noise. The localization errors are depicted in Fig. 10(a) and 10(b), and the rewards during the iteration process are shown in Fig. 10(c) and 10(d). From Fig. 10, it is evident that the position information of the target can be correctly estimated by the proposed estimator, as the localization errors approximately converge to the CRLB while the rewards reach the maximum value. These simulation results confirm the effectiveness of the algorithm.

Fig. 11 shows the RMSE and the BNorm performance of the estimators concerning the measurement noise standard deviation. Obviously, as the error increases, the RMSE values of the PLE, TLS, WIVE and GN algorithms gradually deviate from the CRLB and their bias values increase rapidly with the growing noise. Among all the simulated estimators, PLE exhibits the most significant bias when $\sigma < 4^\circ$, whereas WIVE demonstrates the highest bias for $\sigma > 4^\circ$. This occurs because the effectiveness of the instrumental variable matrix and the data matrix in WIVE can be substantially diminished in the presence of large measurement noise. It is noteworthy that the RMSE performance of GN algorithm exhibits a gradual degradation as the level of noise increases. This phenomenon arises due to the escalating linearization error in GN algorithm. As noise intensifies, the linearization error becomes more pronounced, leading to a continuous rise in bias, as visually depicted in Fig. 11(b). Remarkably, the proposed estimator successfully addresses the inherent linearization limitations observed in the GN algorithm. This is vividly illustrated in Fig. 11, where the estimation error consistently approaches the CRLB across various noise levels. In addition, the proposed algorithm exhibits minimal bias. These performances exemplify the superiority of the proposed algorithm.

2) 3D AOA Localization: Expanding AOA localization from 2D space to 3D space is necessary, as it better aligns with the requirements of real-world scenarios. Our method can be easily modified to enable target positioning in 3D space. Let $\boldsymbol{p} = [x, y, z]^T$ and $\boldsymbol{o}_i = [x_k, y_k, z_k]^T$ denote the target and sensor k positions in 3D space. The sensors provide the target

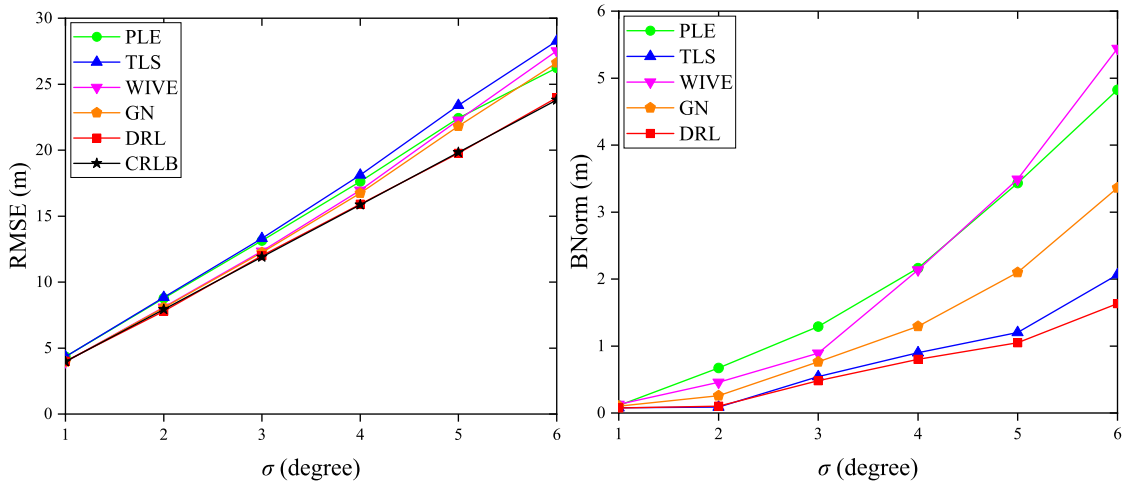


(a) Localization errors for $\sigma = 1° \sim 3°$ (b) Localization errors for $\sigma = 4° \sim 6°$



(c) Episode rewards for $\sigma = 1° \sim 3°$ (d) Episode rewards for $\sigma = 4° \sim 6°$

Fig. 10. 2D AOA localization performance of the proposed MLE.



(a) RMSE against measurement noise standard deviation (b) BNorm against measurement noise standard deviation

Fig. 11. RMSE and BNorm of position estimate for different 2D estimators.

bearing and elevation angles. The measurement model for the elevation angle measurement is given by

$$\tilde{\phi}_k = \phi_k + m_k, \tag{52}$$

where

$$\phi_k = \arctan \frac{z - z_k}{||\boldsymbol{p} - \boldsymbol{o}_k||_2}$$

represents the true elevation angle, and m_k is the measurement noise, assumed to be zero-mean white Gaussian with variance $\sigma^2_{m_k}$.

The elevation and bearing measurement noises are assumed to be independent. Hence, the maximum likelihood estimation for the parameter vector $\boldsymbol{p}$ can be expressed as:

$$\hat{\boldsymbol{p}}_{\text{MLE}} = \arg \min_{\boldsymbol{p} \in \mathbb{R}^3} \frac{1}{2} (\tilde{\boldsymbol{z}} - \boldsymbol{z}(\boldsymbol{p}))^T \boldsymbol{R}^{-1} (\tilde{\boldsymbol{z}} - \boldsymbol{z}(\boldsymbol{p})), \tag{53}$$

where

$$\tilde{\boldsymbol{z}} = [\tilde{\theta}_1, \tilde{\theta}_2, \ldots, \tilde{\theta}_N, \tilde{\phi}_1, \tilde{\phi}_2, \ldots, \tilde{\phi}_N]^T$$

denotes the $2N \times 1$ vector of noisy bearing and elevation measurements,

$$\boldsymbol{z} = [\theta_1, \theta_2, \ldots, \theta_N, \phi_1, \phi_2, \ldots, \phi_N]^T$$

represents the $2N \times 1$ vector of true bearing and elevation angles, and

$$\boldsymbol{R} = diag(\sigma^2_{n_1}, \sigma^2_{n_2}, \ldots, \sigma^2_{n_N}, \sigma^2_{m_1}, \sigma^2_{m_2}, \ldots, \sigma^2_{m_N})$$

denotes the $2N \times 2N$ diagonal covariance matrix of the angle noise. In terms of (11)-(14), the fundamental components of the MDP for the 3D AOA localization problem can be defined as $s_i \triangleq (\hat{\boldsymbol{p}}_i, \tilde{\boldsymbol{z}})$, $a_i \triangleq \boldsymbol{\zeta}_i$, and $r_i \triangleq -\frac{1}{2}(\tilde{\boldsymbol{z}} - \boldsymbol{z}(\hat{\boldsymbol{p}}_i))^T \boldsymbol{R}^{-1} (\tilde{\boldsymbol{z}} - \boldsymbol{z}(\hat{\boldsymbol{p}}_i))$.

Fig. 12 illustrates the simulated geometry for 3D AOA localization. The true target is positioned at (230m, 80m, 125m). Four sensors with coordinates (0m, 0m, 0m), (0m, 0m, 100m), (0, 100m, 0m), and (0, 100m, 100m) are utilized to locate the target. The noise in bearing and elevation angle measurements is assumed to be i.i.d. with $\sigma^2_{n_1} = \cdots = \sigma^2_{n_N} = \sigma^2_{m_1} = \cdots = \sigma^2_{m_N} = \sigma^2$.

Fig. 13 illustrates the learning curves of the proposed estimator for various values of σ, ranging from 0.5° to 3°. It is evident that the proposed estimator demonstrates convergence in 3D AOA localization across diverse noise levels. Fig. 14(a)–14(f) present the exploration trajectories and the corresponding estimated trajectories. To evaluate the localization accuracy throughout the iteration process, the localization error of the estimator at iteration step k is defined as $error_k = \sqrt{(\hat{x}_k - x)^2 + (\hat{y}_k - y)^2 + (\hat{z}_k - z)^2}$. The localization errors are depicted in Fig. 15(a) and 15(b), while the rewards during the iteration process are shown in Fig. 15(c) and 15(d). From Fig. 15(a)–15(d), it can be observed that the position information of the target can be accurately estimated, as the localization errors approximately converge to the CRLB and the reward reaches its peak value for various noise standard deviation. These simulation results confirm the effectiveness of the algorithm in 3D AOA localization.

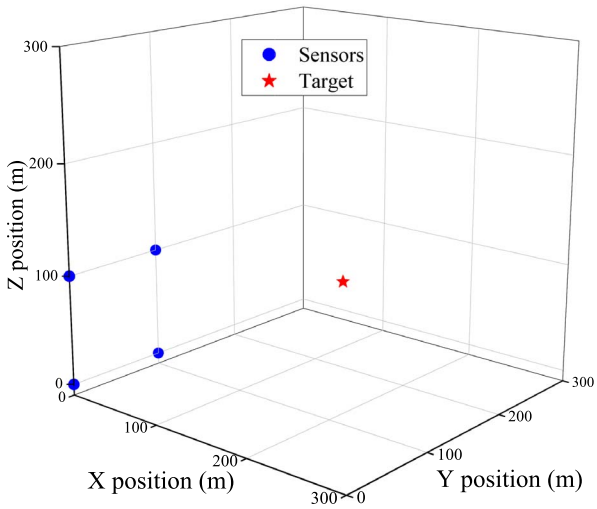
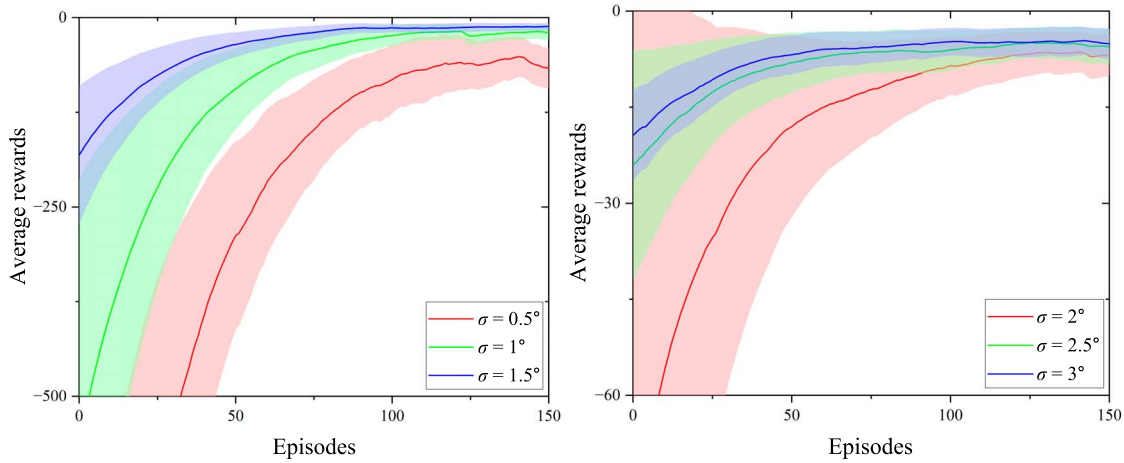


Fig. 12. 3D simulated AOA localization geometry.

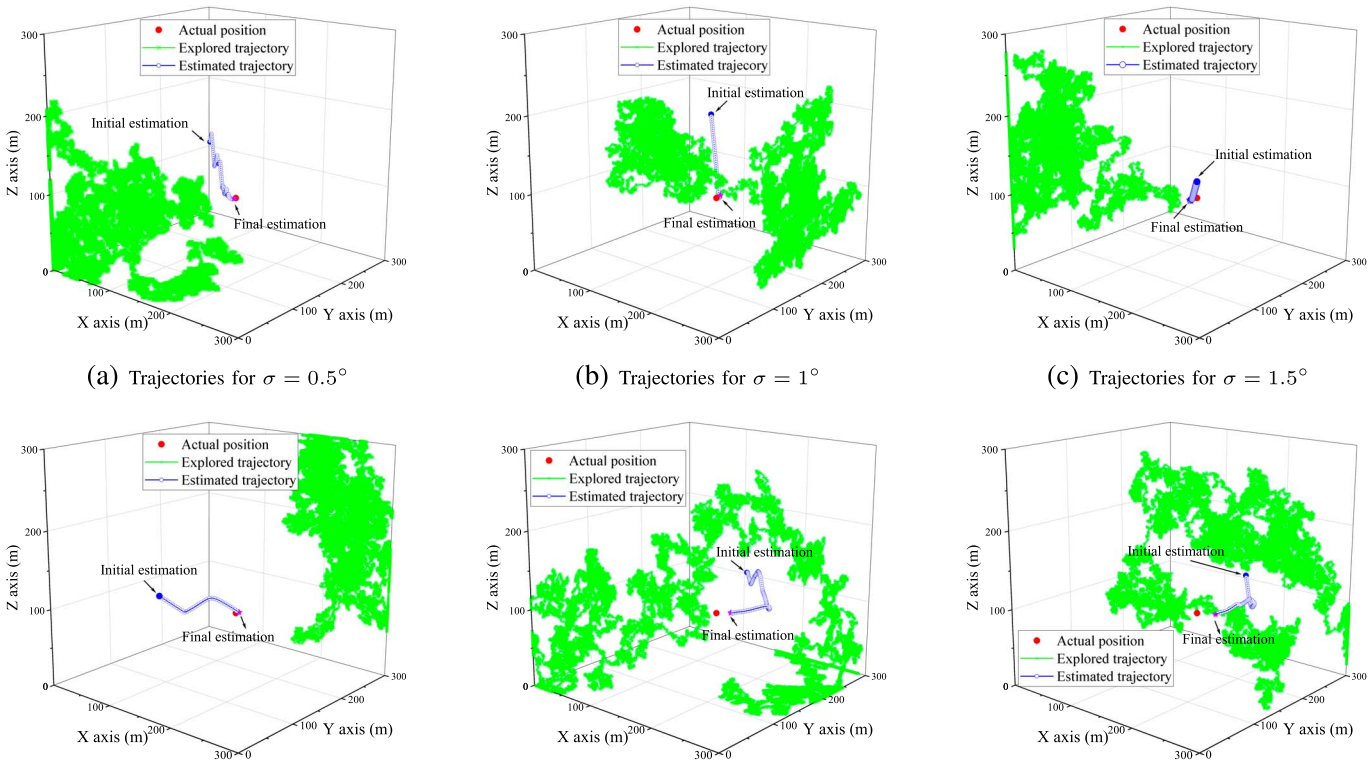
The simulation results for the RMSE and BNorm of the estimators with respect to the noise standard deviation σ are shown in Fig. 16. As it can be seen, the OVE has the largest RMSE compared to all other estimators. For $\sigma \leq 1.5^\circ$, the 3D-WIVE and 3D-PLE algorithms show similar RMSE performance. However, when $\sigma > 1.5^\circ$, the localization error of the 3D-WIVE algorithm becomes greater than that of 3D-PLE. This is similar to the 2D localization case, and is due to the degradation of the instrumental variable matrix and data matrix with increasing noise. This observation is also evident from Fig. 16(b), where with increasing noise, the BNorm of 3D-WIVE gradually exceeds that of 3D-PLE. It is interesting to note that the GN, CIM and the proposed DRL-based estimator achieve optimal RMSE performance by closely approximating the CRLB across various values of σ. However, as depicted in Fig. 16(b), the BNorm of CIM surpasses that of GN and DRL. Moreover, for noise levels $\sigma \leq 2^\circ$, GN and DRL exhibit similar BNorm values; nevertheless, with increasing noise, the first-order linearization error of the GN algorithm gradually increases, resulting in its BNorm surpassing that of DRL when $\sigma \geq 2^\circ$. Consequently, these results highlight the superiority of the proposed algorithm in maintaining reliable and precise estimation even in the presence of escalating noise levels.

3) Running time analysis: Table II presents the running times for both the proposed DRL-based MLE and baseline estimators applied for 2D and 3D AOA localization. These results were averaged over 10,000 rounds and computed using a CPU i5-7400. For the iterative estimators GN, DRL and CIM, the number of iterations was set to 10. As can be seen from the table, in 2D localization estimation, the iterative estimators GN and DRL have higher computational complexity compared to the analytical estimators PLE, TLS and WIVE, since their complexity depends on the number of iterations. Among these, DRL has the highest computation complexity due to its relation



(a) Learning curves for $\sigma = 0.5^\circ \sim 1.5^\circ$ (b) Learning curves for $\sigma = 2^\circ \sim 3^\circ$

Fig. 13. Effectiveness of the proposed DRL-based MLE for 3D AOA localization.

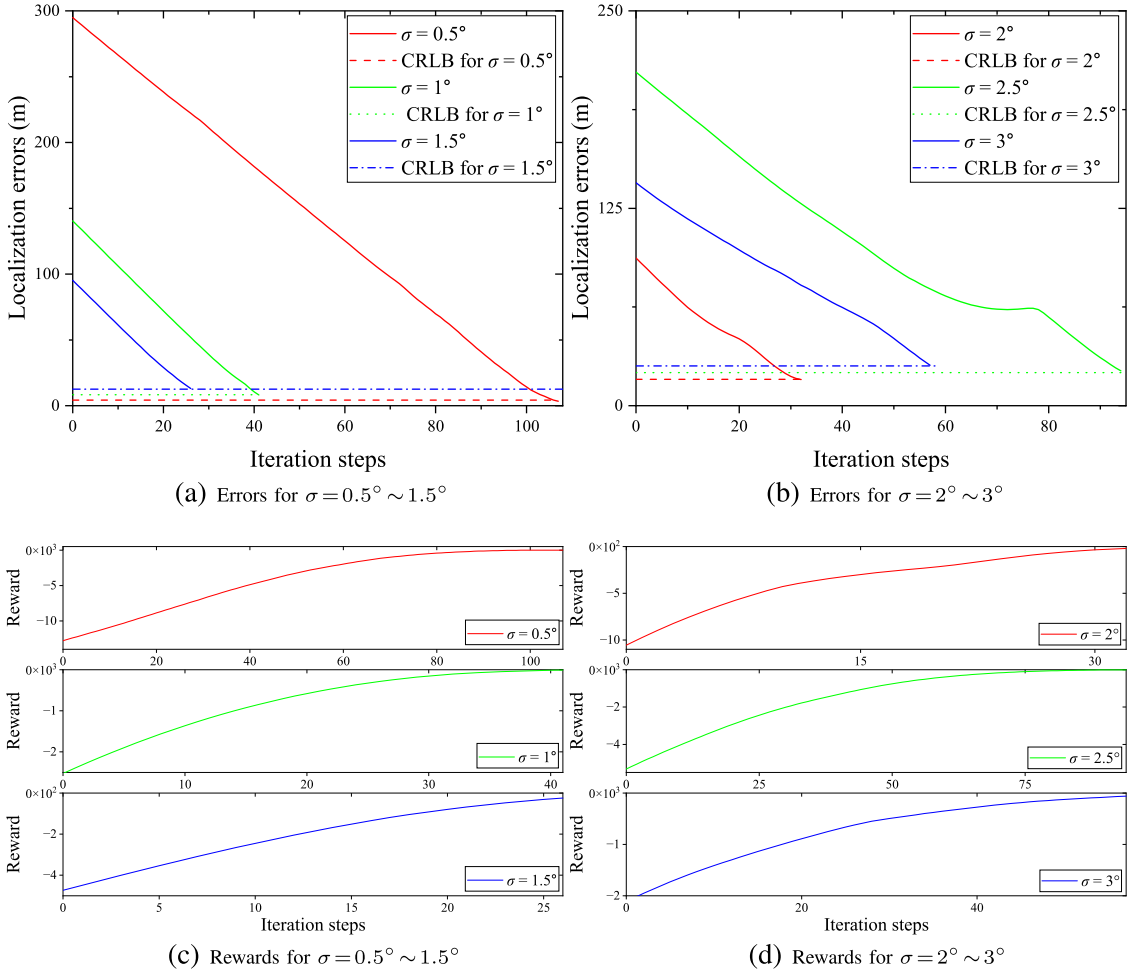


(a) Trajectories for $\sigma = 0.5^\circ$ (b) Trajectories for $\sigma = 1^\circ$ (c) Trajectories for $\sigma = 1.5^\circ$

(d) Trajectories for $\sigma = 2^\circ$ (e) Trajectories for $\sigma = 2.5^\circ$ (f) Trajectories for $\sigma = 3^\circ$

Fig. 14. Description of agent and environment interaction in 3D space.

to the number of layers and neurons in the neural network. Additionally, WIVE has higher complexity than PLE as it requires constructing the weight matrix using PLE. In 3D localization estimation, similar to 2D, the running times of the iterative estimators GN, CIM and DRL are higher than OVE, 3D-PLE and 3D-WIVE. Interestingly, the running time of GN is higher than DRL in this case, because the initialization of GN comes from 3D-PLE, which has higher complexity than PLE. It is noteworthy that although CIM has a shorter running time among iterative estimators, as noted in literature [6], it requires massive iterations to converge, so its running time would be much higher when convergence is reached. Overall, the computation time of the proposed DRL-based MLE is within an acceptable range.



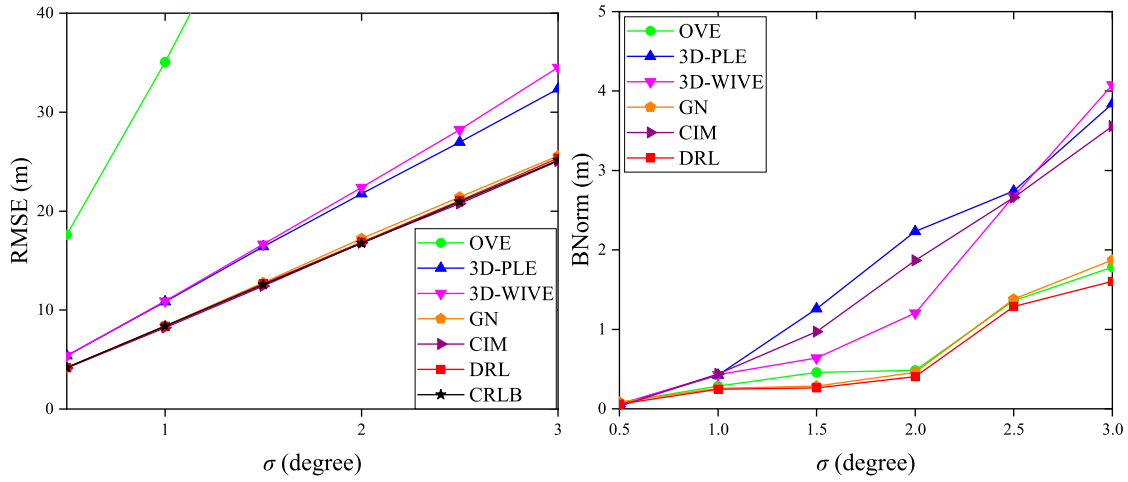
(a) Errors for $\sigma = 0.5^\circ \sim 1.5^\circ$

(b) Errors for $\sigma = 2^\circ \sim 3^\circ$

(c) Rewards for $\sigma = 0.5^\circ \sim 1.5^\circ$

(d) Rewards for $\sigma = 2^\circ \sim 3^\circ$

Fig. 15. 3D AOA localization performance of the proposed MLE.



(a) RMSE against measurement noise standard deviation

(b) BNorm against measurement noise standard deviation

Fig. 16. RMSE and BNorm of position estimate for different 3D estimators.

TABLE II
COMPARISON OF RUNNING TIMES

Algorithms	Running Time (ms)	Algorithms	Running Time (ms)
PLE	0.1219	OVE	0.1631
TLS	0.1079	3D-PLE	0.1983
WIVE	0.3256	3D-WIVE	0.4303
GN	1.5109	GN	3.1182
DRL	2.6210	CIM	0.9669
-	-	DRL	2.6654

V. CONCLUSION

This paper presents a learning-based MLE for AOA localiza tion. The proposed approach incorporates deep reinforcement learning to address the convergence issues typically encountered in traditional MLEs. To assess its performance, we compare the learning-based MLE with other state-of-the-art estimators in both 2D and 3D AOA localization. Through simulations, we consistently observe that the proposed learning-based MLE performs excellently in both 2D and 3D AOA localization scenarios.

REFERENCES

[1] A. N. Bishop, B. D. O. Anderson, B. Fidan, P. N. Pathirana, and G. Mao, "Bearing-only localization using geometrically constrained optimization," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 1, pp. 308–320, Jan. 2009.

[2] Y. Wang and K. C. Ho, "Unified near-field and far-field localization for AOA and hybrid AOA-TDOA positionings," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1242–1254, Feb. 2018.

[3] T. Northardt and S. C. Nardone, "Track-before-detect bearings-only localization performance in complex passive sonar scenarios: A case study," *IEEE J. Ocean. Eng.*, vol. 44, no. 2, pp. 482–491, Apr. 2019.

[4] J. A. Luo, X. H. Shao, D. L. Peng, and X. P. Zhang, "A novel subspace approach for bearing-only target localization," *IEEE Sens. J.*, vol. 19, no. 18, pp. 8174–8182, Sep. 2019.

[5] N. H. Nguyen, K. Doğançay, and E. E. Kuruoglu, "An iteratively reweighted instrumental-variable estimator for robust 3-D AOA localization in impulsive noise," *IEEE Trans. Signal Process.*, vol. 67, no. 18, pp. 4795–4808, Sep. 2019.

[6] Y. Zou, L. Wu, J. Fan, and H. Liu, "A convergent iteration method for 3-D AOA localization," *IEEE Trans. Veh. Technol.*, vol. 72, no. 6, pp. 8267–8271, Jun. 2023, doi:1003609910.1109/TVT.2023.3242054.

[7] R. G. Stansfield, "Statistical theory of DF fixing," *J. Inst. Elect. Eng.—Part IIIA, Radiocommun*., vol. 94, no. 15, pp. 762–770, Dec. 1947.

[8] S. Nardone, A. Lindgren, and K. Gong, "Fundamental properties and performance of conventional bearings-only target motion analysis," *IEEE Trans. Autom. Control*, vol. 29, no. 9, pp. 775–787, Sep. 1984.

[9] K. Doğançay, "On the bias of linear least squares algorithms for passive target localization," *Signal Process.*, vol. 84, no. 3, pp. 475–486, Mar. 2004.

[10] Y. T. Chan and S. W. Rudnicki, "Bearings-only and Doppler-bearing tracking using instrumental variables," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 4, pp. 1076–1083, Oct. 1992.

[11] K. Doğançay, "Passive emitter localization using weighted instrumental variables," *Signal Process.*, vol. 84, no. 3, pp. 487–497, 2004.

[12] K. Doğançay, "Bearings-only target localization using total least squares," *Signal Process.*, vol. 85, no. 9, pp. 1695–1710, 2005.

[13] K. Doğançay, and G. Ibal, "3D passive localization in the presence of large bearing noise," in *Proc. 13th Eur. Signal Process. Conf.,* Antalya, Turkey, Sep. 2005.

[14] Y. Wang and K. C. Ho, "An asymptotically efficient estimator in closedform for 3D AOA localization using a sensor network," *IEEE Trans. Wireless Commun.*, vol. 14, no. 12, pp. 6524–6535, Dec. 2015.

[15] N. Adib, S. C. Douglas, "Extending the Stansfield algorithm to three dimensions: algorithms and implementations," *IEEE Trans. Signal Process.*, vol. 66, no.4, pp.1106–1117, Feb. 2018.

[16] S. C. Nardone, A. G. Lindgren, and K. F. Gong, "Fundamental properties and performance of conventional bearings-only target motion analysis," *IEEE Trans. Autom. Control*, vol. AC-29, no. 9, pp. 775–787, Sep. 1984.

[17] M. A. Doron, A. J. Weiss, and H. Messer, "Maximum-likelihood direction finding of wide-band sources," *IEEE Trans. Signal Process.*, vol. 41, no. 1, pp. 411–414, Jan. 1993.

[18] W. H. Foy, "Position-location solution by Taylor-series estimations," *IEEE Trans. Aerosp. Electron. Syst.,* vol. AES-12, no. 2, pp. 187–194, Mar. 1976.

[19] Z. Wang, J.-A. Luo, and X.-P. Zhang, "A novel location-penalized maximum likelihood estimator for bearing-only target localization," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6166–6181, Dec. 2012.

[20] Y. T. Chan, H. Y. C. Hang, and P.-C. Ching, "Exact and approximate maximum likelihood localization algorithms," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 10–16, Jan. 2006.

[21] M. Gavish and A. J. Weiss, "Performance analysis of bearing-only target location algorithm," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 26, pp. 22–26, Jul. 1992.

[22] A. M. Ahmed, A. A. Ahmad, S. Fortunati, A. Sezgin, M. S. Greco, and F. Gini, "A reinforcement learning based approach for multitarget detection in massive MIMO radar," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 2622–2636, Oct. 2021.

[23] F. Meng, K. Tian, and C. Wu, "Deep reinforcement learning-based radar network target assignment," *IEEE Sens. J.*, vol. 21, no. 14, pp. 16315–16327, Jul. 2021.

[24] S. Z. Gurbuz, I. Bilik, and L. Rosenberg, "Guest editorial for the TAES special section on deep learning for radar applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 60, no. 1, pp. 4–7, Feb. 2024.

[25] P. Lang et al., "A comprehensive survey of machine learning applied to radar signal processing," 2020, *arXiv:2009.13702*.

[26] S. Li, G. Liu, K. Zhang, Z. Qian, and S. Ding, "DRL-based joint path planning and jamming power allocation optimization for suppressing netted radar system," *IEEE Signal Process. Lett.,* vol. 30, pp. 548–552, 2023.

[27] C. E. Thornton, M. A. Kozy, R. M. Buehrer, A. F. Martone, and K. D. Sherbondy, "Deep reinforcement learning control for radar detection and tracking in congested spectral environments," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 4, pp. 1335–1349, Dec. 2020.

[28] Z. Shan, P. Liu, L. Wang L, and Y. Liu. "A cognitive multi-carrier radar for communication interference avoidance via deep reinforcement learning," *IEEE Trans. Cogn. Commun. Netw.*, vol. 9, no. 6, pp. 1561–1578, Dec. 2023.

[29] X. Jiang, T. Liu, Y. Liu, S. Zhang, H. Lin, and L. Liu, "An azimuth aware deep reinforcement learning framework for active SAR target recognition," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 4936–4951, 2024.

[30] K. Li, B. Jiu, P. Wang, H. Liu, and Y. Shi, "Radar active antagonism through deep reinforcement learning: A way to address the challenge of mainlobe jamming," *Signal Process.*, vol. 186, 2021, Art. no. 108130.

[31] Z. Pan, Y. Li, S. Wang, and Y. Li, "Joint optimization of jamming type selection and power control for countering multifunction radar based on deep reinforcement learning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 4, pp. 4651–4665, Aug. 2023.

[32] F. Pang, K. Doğançay, N. H. Nguyen, and Q. Zhang, "AOA pseudolinear target motion analysis in the presence of sensor location errors," *IEEE Trans. Signal Process.*, vol. 68, pp. 3385–3399, 2020.

[33] H. Dong, H. Dong, Z. Ding, S. Zhang, and Chang, *Deep Reinforcement Learning*, 1st ed. Singapore: Springer, 2020.

[34] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015.

[35] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[36] B. D. Ziebart et al., "Maximum entropy inverse reinforcement learning," in *Proc. Assoc. Adv. Artif. Intell.*, 2008, pp. 1433–1438.

[37] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.,* vol. 80, 2018, pp. 1861–1870.

[38] T. Haarnoja et al., "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.

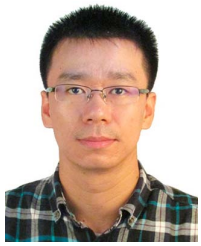
[39] H. V. Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, Feb. 2016, pp. 2094–2100.

[40] K. M. Steven, *Fundamentals of Statistical Signal Processing: Estimation Theory*. Upper Saddle River, NJ, USA: Prentice-Hall, 1993.
[41] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, "Intelligent resource scheduling for 5G radio access network slicing," *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7691–7703, Aug. 2019.
[42] F. Fu, Y. Kang, Z. Zhang, F. R. Yu, and T. Wu, "Soft actor–critic DRL for live transcoding and streaming in vehicular fog-computing-enabled IoV," *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1308–1321, Feb. 2021.
[43] T. P. Lillicrap et al., "Continuous control with deep reinforcement learning," in *Proc. Int. Conf. Learn. Representations*, 2016.
[44] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. Int. Conf. Mach. Learn.,* 2018, pp. 1587–1596.
[45] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn. PMLR*, 2015, pp. 1889–1897.
[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[47] K. Doğançay and H. Hmam, "Optimal angular sensor separation for AOA localization," *Signal Process.*, vol. 88, no. 5, pp. 1248–1260, May 2008.
[48] A. N. Bishop, B. Fidan, B. Anderson, K. Doğançay, and P. N. Pathirana, "Optimality analysis of sensor-target localization geometries," *Automatica*, vol. 46, no. 3, pp. 479–492, 2010.
[49] S. Xu and K. Doğançay, "Optimal sensor placement for 3-D angle-of-arrival target localization," *IEEE Trans. Aerosp. Electron. Syst.,* vol. 53, no. 3, pp. 1196–1211, Jun. 2017.
[50] L. Graesser and W. L. Keng, *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*. London, U.K.: Pearson, 2020.
[51] F. Xu, Y. Ruan and Y. Li, "Soft actor–critic based 3-D deployment and power allocation in cell-free unmanned aerial vehicle networks," *IEEE Wireless Commun. Lett.,* vol. 12, no. 10, pp. 1692–1696, Oct. 2023.

Chengyi Zhou is currently working toward the Ph.D. degree with the College of Electrical Engineering, Zhejiang University, Hangzhou, China, under the supervision of Prof. M. Liu. His research interests include machine learning, deep reinforcement learning, and their applications in target localization and tracking.

Meiqin Liu (Senior Member, IEEE) received the B.E. and Ph.D. degrees in control theory and control engineering from the Central South University, Changsha, China, in 1994 and 1999, respectively. She was a Postdoctoral Research Fellow with the Huazhong University of Science and Technology, Wuhan, China, from 1999 to 2001. She was a Visiting Scholar with the University of New Orleans, New Orleans, LA, USA, from 2008 to 2009. She was a Professor with the College of Electrical Engineering, Zhejiang University, Hangzhou, China, from 2001 to 2021. She is currently a Professor with the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Xi'an Jiaotong University, Xi'an, China, and also with the College of Electrical Engineering, Zhejiang University, Hangzhou, China. She has authored more than 200 papers in major journals and international conferences. She has led 16 national or provincial or ministerial projects in the last five years, including nine projects funded by the National Natural Science Foundation of China (NSFC). Her work was supported by Zhejiang Provincial Natural Science Fund for Distinguished Young Scholars in 2010, and by the National Science Fund for Excellent Young Scholars of China in 2012. She has won a second prize of Science and Technology Award of Zhejiang Province in 2013, and a first prize of Natural Science Award of Chinese Association of Automation in 2019, respectively. Her research interests include theory and application of artificial intelligence, multi-sensor networks, information fusion, and nonlinear systems.

Senlin Zhang (Member, IEEE) received the B.E. degree from Wuhan University of Technology, Wuhan, China, in 1984, and the M.E. degree in control theory and control engineering from Zhejiang University, Hangzhou, China, in 1991, both in control theory and control engineering. He is currently a Professor with the College of Electrical Engineering and the National Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, China. He has authored more than 200 papers in major journals and international conferences. His current research interests include unmanned undersea systems, intelligent systems, and multi-sensor networks.



Ronghao Zheng (Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in control theory and control engineering from Zhejiang University, Hangzhou, China, in 2007 and 2010, respectively, and the Ph.D. degree in mechanical and biomedical engineering from the City University of Hong Kong, in 2014. He is currently with the College of Electrical Engineering, Zhejiang University. His research interests include distributed algorithms and control, especially the coordination of networked mobile robot teams with applications in automated systems and security.

Shanling Dong (Member, IEEE) received the B.S. degree in automation from Xidian University, Xi'an, China, in 2014, and the Ph.D. degree in Control Science and Engineering from Zhejiang University, Hangzhou, China, in 2019. Supported by China Scholarship Council, she was a Visiting Graduate Student with the University of California, Rriverside, USA, from 2017 to 2018. She was a Research Associate and Postdoctoral Research Fellow with the City University of Hong Kong, Hong Kong SAR, China, from 2019 to 2020. She is currently with the College of Electrical Engineering, Zhejiang University, Hangzhou, China. Her research interests include Markov jump systems, fuzzy systems, multiagent systems, and ocean robots.

Zhunga Liu (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees from the Northwestern Polytechnical University (NPU), Xi'an, China, in 2007, 2010, and 2013 respectively. He also studied in Telecom Bretagne, France for Ph.D. during 2010 and 2013. He has been a Professor with the School of Automation, NPU, since 2017. His research interests include information fusion and pattern recognition.