



Article

Data-Driven Joint Beam Selection and Power Allocation for Multiple Target Tracking

Yuchun Shi ^{1,*}, Hao Zheng ² and Kang Li ²¹ School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, China² National Laboratory of Radar Signal Processing, Xidian University, Xi'an 710071, China; hzheng1994@stu.xidian.edu.cn (H.Z.); kli_6@stu.xidian.edu.cn (K.L.)

* Correspondence: 20211048@zknknu.edu.cn

Abstract: For the problem of joint beam selection and power allocation (JBSPA) for multiple target tracking (MTT), existing works tend to allocate resources only considering the MTT performance at the current tracking time instant. However, in this way, it cannot guarantee the long-term MTT performance in the future. If the JBSPA not only considers the tracking performance at the current tracking time instant but also at the future tracking time instant, the allocation results are theoretically able to enhance the long-term tracking performance and the robustness of tracking. Motivated by this, the JBSPA is formulated as a model-free Markov decision process (MDP) problem, and solved with a data-driven method in this article, i.e., deep reinforcement learning (DRL). With DRL, the optimal policy is given by learning from the massive interacting data of the DRL agent and environment. In addition, in order to ensure the information prediction performance of target state in maneuvering target scenarios, a data-driven method is developed based on Long-short term memory (LSTM) incorporating the Gaussian mixture model (GMM), which is called LSTM-GMM for short. This method can realize the state prediction by learning the regularity of nonlinear state transitions of maneuvering targets, where the GMM is used to describe the target motion uncertainty in LSTM. Simulation results have shown the effectiveness of the proposed method.



Citation: Shi, Y.; Zheng, H.; Li, K. Data-Driven Joint Beam Selection and Power Allocation for Multiple Target Tracking. *Remote Sens.* **2022**, *14*, 1674. <https://doi.org/10.3390/rs14071674>

Academic Editor: Andrzej Stateczny

Received: 8 March 2022

Accepted: 29 March 2022

Published: 30 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: multiple target tracking; joint beam selection and power allocation; deep reinforcement learning

1. Introduction

Multiple target tracking (MTT) is an important research topic in the military field, and also a difficult problem at present [1–3]. Multiple-input multiple-output (MIMO) [4] radar is an extension of traditional phased array radar, and it is able to transmit signals in all directions and multi-beams at the same time. This operating mode is called simultaneous multi-beam mode [5]. Technically speaking, in the simultaneous multi-beam mode, a single collocated MIMO radar can track multiple targets simultaneously by controlling different beams tracking different targets independently. In this case, the complex data association process of tracking can be avoided since the spatial diversity of multiple targets can be used to distinguish the echoes of different targets, and thus, the multi-target tracking problem can be divided into multiple single-target tracking problems.

In practical applications, we have to consider two limiting factors inherent to collocated MIMO radar [6]. (1) The maximum number of beams that the system can generate is limited at each time. In theory, if the number of MIMO radar transmitters is N , the system can only generate M ($M < N$) orthogonal beams at most at each time. (2) The total transmitted power of simultaneous multiple beams is limited by the hardware. In real applications, it is necessary to limit the total transmitted power of multiple beams to ensure that the total transmission power of the system does not exceed the tolerance of the hardware. With the traditional transmitting mode, the number of beams is always set to a constant, and the

transmitting power is uniformly allocated to the beams. Although it is easy to implement in engineering, the utilization efficiency of the resource is relatively low. In terms of the kinds of resources, the existing resource allocation methods can be roughly divided into two groups: one is based on the transmitting parameters, the other is based on the system composition structure. In this paper, the discussed problem of power and beam resource allocation belongs to the first one. The resource allocation based on the system composition structure mostly exists in multiple radar systems (MRS).

In [7], the authors used the covariance matrix of the tracking filter as the evaluation of tracking performance, and established and solved an optimization cost function of transmitted power. The authors of [8] proposed a cognitive MTT method based on prior information, and designed a cognitive target tracking framework. With this method, the radar transmit power parameters can be adaptively adjusted at each tracking time based on the information fed back by the tracker. In [9], an MRS resource allocation method is proposed for MTT based on quality of service constrained (QoS). This method cannot only illuminate multiple targets in groups, but also optimize the transmitted power of each radar. The authors of [10–14] also studied the optimization-based resource allocation method with different resources limitation in different target–radar scenarios. Their similarity is to take the localization Cramér–Rao lower bound (CRLB) or the tracking Bayesian Cramér–Rao lower bound (BCRLB) [13] as the performance evaluation in the cost function. However, the used CRLBs and BCRLBs are both derived with an assumption of constant velocity (CV) model or constant acceleration (CA) model, we thus conclude that these kinds of methods are model-based methods. These model-based methods can achieve excellent results on the premise that the target’s motion is conform to the model assumption. However, in the case of maneuvering targets (the targets undergoing non-uniform-linear motions with certain uncertainty and randomness), they would not work well because the inevitable model mismatch causes a lower resource utilization, a larger prediction error of the target state, and an unreliable BCRLB. To overcome this, in [15], the resource problem was first formulated as a Markov decision process (MDP) and solved by a data-driven method, where recurrent neural network (RNN) [16,17] was used to learn a prediction model for BCRLB and target’s state, and deep-reinforcement-learning (DRL) [18,19] was adopted for solving the optimal power allocation solution. This method works under the condition that the number of generated beams is equal to the number of targets. However, in practice, radar system always have to face the challenge that the maximum beam number is far less than the number of targets. On this basis, the authors further propose a constrained-DRL-based MRS resource allocation method for meeting the given accuracy of single target tracking [20]. It is noticed that, in [15,20], the output of RNN is characterized as a single Gaussian distribution. In this way, an accurate predictive information can be obtained as the target is maneuvering with low uncertainty. Based on above studies, in this paper, we will further discuss how to jointly schedule the power and beam resources of a radar system to enhance the overall performance of MTT in maneuvering target scenarios.

In this paper, a novel data-driven joint beam selection and power allocation (JBSPA) method is introduced. The main contributions of this paper can be summarized as follows:

- (1) A JBSPA method for MTT is developed based on deep learning, which aims to maximize the performance of MTT by adjusting power and beam allocation within the limited resources at each illumination. Through mathematical optimization modeling and analysis, the problem of JBSPA is essentially a non-convex and nonlinear problem, and the conventional optimization method is thus not suitable. To address this problem, we characterize the original optimization cost function as a model-free MDP problem by designing and introducing a series of variable parameters of MDP, including state, action and reward. For a discrete MDP, a common way is dynamic programming (DP). However, our problem is a continuous controlling problem over the action and state space, and thus, DP-based methods are not available. Since the emerging DRL is a good model-free MDP [21], it will be taken to solve MDP of JBSPA (DRL-JBSPA) in this paper.

- (2) Considering the possible existence of a model-mismatch problem in maneuvering target tracking, the long short-term memory (LSTM) [17] incorporating the Gaussian mixture model (GMM), say, LSTM-GMM, is used to derive prior information representation of targets for resource allocation. The biggest change exists in the output layer of LSTM; there, two different GMMs are utilized to describe the uncertainty of state transition of the maneuvering target and radar cross-section (RCS), respectively. In this way, at each tracking interval, the LSTM-GMM will not output a determined predicted extended target state, but multiple Gaussian distributions. Then, we select one of them randomly according to mixed coefficients, to obtain the predicted target state and RCS, and the Bayesian information matrix (BIM) of prior information at the next tracking time instant.
- (3) Considering that the JBSPA is a continuous control problem over state and action spaces, in order to improve the computational efficiency, this work adopts a deterministic policy function in calculation of Q function, and deduces the deterministic policy gradient [22,23] by deriving the Q function with respect to policy parameters. Then, we apply an actor network and a critic network to enhance the stability of the training. A modified actor network is designed to improve the applicability of the original actor-critic frame to JBSPA, where the output transmit parameters (i.e., the number of generated beams by the radar, the pointing of beams, and the power of each beam) are realized by taking three different output layers into the action network. Due to the usage of a trained network and avoiding optimization solving at each tracking time, the requirement of real time in resource allocation can be guaranteed in MTT.

The outline of the paper is as follows. In Section 2, a brief review of DRL is introduced. In Section 3, the resource allocation problem is described. In Section 3, the resource allocation problem is abstracted into a mathematical cost function. In Section 4, a deep-learning-based JBSPA method is detailed. In Section 5, several numerical experiment results are provided to verify the effectiveness of the proposed method. Some problems and future works are discussed in Section 6. In Section 7, the conclusion of this paper is given.

2. Markov Decision Process and Reinforcement Learning

In this section, a brief introduction MDP and reinforcement learning (RL) is given. Being the fundamental model, MDP is often used to describe and solve the sequential decision-making problems involving uncertainties. In general, a model-free MDP is described as a tuple $(\mathbf{S}, \mathbf{A}, r, \gamma)$, where $\mathbf{S}$ is the state set in the decision process, $\mathbf{A}$ is the action set, r is the reward feedback by the environment after executing a certain action, and γ is the discounted factor for measuring the current value of future return. The aim of MDP is to find the optimal policy that can maximize the cumulative discounted reward $\sum_{k=0}^{\infty} \gamma^k r_{t+k}$, and the policy function $\pi(\mathbf{a}|\mathbf{s})$ is expressed as follows:

$$\pi(\mathbf{a}|\mathbf{s}) = p(\mathbf{A}_t = \mathbf{a}|\mathbf{S}_t = \mathbf{s}) \quad (1)$$

which is the probability of taking action $\mathbf{a}$ with current state $\mathbf{s}$.

Due to the randomness of the policy, the expectation operation is used to calculate the cumulative discounted reward function. When an action is given under a certain state, the cumulative discounted reward function becomes the state-action value function, i.e.,

$$Q(\mathbf{s}, \mathbf{a}) = \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k} | \mathbf{S}_t = \mathbf{s}, \mathbf{A}_t = \mathbf{a} \right] \quad (2)$$

which is also called Q function. This function is generally used in DRL for seeking the optimal policy.

In reinforcement learning (RL), there is an agent that can interact with the environment by actions, accumulate interactive data, and learn optimal action policy. In general, the commonly used reinforcement learning methods are classed into two kinds: one is based

on the value function and the other is based on the policy gradient (PG). In value-function-based RLs, such as Q-learning, the learning criteria are looking for the action that can maximize the Q function by exploring the whole action space, i.e.,

$$\mathbf{a}^* = \arg \max_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a}) \quad (3)$$

For the applications on continuous state and action space, value-function-based methods are not always available. Different from the circuitous way of Q-learning enumerating all action-value functions, policy-gradient-based methods find the optimal policy by directly calculating the update direction of policy parameters [21]. Therefore, the PG is more suitable for MDP problems over continuous action and state space. The goal of PG is to find the policy π_{θ}^* that can maximize the expectation of cumulative discounted reward, i.e.,

$$\pi_{\theta}^* = \arg \max_{\pi_{\theta}} \mathbb{E}_{\tilde{\tau} \sim \pi} [G(\tilde{\tau})] \quad (4)$$

where $\tilde{\tau} : \mathbf{S}_1, \mathbf{A}_1, \mathbf{S}_2, \mathbf{A}_2, \mathbf{S}_3, \mathbf{A}_3, \mathbf{S}_4, \mathbf{A}_4, \dots$ represents the trajectory obtained by interacting with the environment using policy π_{θ}^* , and $G(\tilde{\tau})$ denotes the cumulative discounted reward on $\tilde{\tau}$. The loss function of the policy is defined as the expectation of the cumulative return, which is given by

$$J(\theta) = \mathbb{E}_{\tilde{\tau} \sim \pi} [G(\tilde{\tau})] = \int_{\tilde{\tau} \sim \pi_{\theta}} \pi_{\theta}(\tilde{\tau}) G(\tilde{\tau}) d\tilde{\tau} \quad (5)$$

where $\pi_{\theta}(\tilde{\tau})$ represents the occurrence probability of the trajectory $\tilde{\tau}$ with the policy π_{θ} . The policy gradient can be obtained by deriving (5), as shown in the following:

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{\tilde{\tau} \sim \pi} [\nabla_{\theta} \log \pi_{\theta}(\tilde{\tau}) G(\tilde{\tau})] \quad (6)$$

Compared with value-function-based methods, the PG-based methods has two advantages, one is higher efficiency in the applications of continuous action and state space, the other is that randomization of the policy can be realized.

3. Problem Statement

There are a monostatic MIMO radar and Q independent point targets in the MTT scenario. The monostatic MIMO radar performs both the transmitting task and receiving task. Assume that the location of MIMO radar is (x_r, y_r) , and the targets is initially located at $(x_0^q, y_0^q), q = 1, 2, 3, \dots, Q$. At k -th tracking time step, the q -th target's state is given by $\mathbf{x}_k^q = [x_k^q, \dot{x}_k^q, y_k^q, \dot{y}_k^q]^T$, in which $[x_k^q, y_k^q]$ and $[\dot{x}_k^q, \dot{y}_k^q]$ denote the corresponding location and speed of q -th target, respectively. Besides that, the RCS components of targets can be also received by the radar at each time step, and that of the q -th target is denoted by $\psi_k^q = [h_{q,k}^R, h_{q,k}^I]^T$, $h_{q,k}^R$ and $h_{q,k}^I$ are the RCS components of channel R and channel I, respectively. If we synthesize the target state component and the RCS component into a vector, an extended state vector can be obtained, i.e., $\zeta_k^q = (\mathbf{x}_k^q, \psi_k^q)^T$.

3.1. Derivation of BCRLB Matrix on Transmitting Parameters

Before modeling JBSPA, we first deduce the mathematical relationship between the tracking BCRLB and the beam selection parameters and power parameters in this section. According to [15], with the assumption that the beam number is the same as the target number, the expression of BIM is

$$J_m(\zeta_k^q) = \Sigma_{\zeta_k^q}^{-1} + \mathbb{E}_{\zeta_k^q} \left[\mathbf{D}_{q,k}^T \Sigma_{\mathbf{y}_k^q}^{-1} \mathbf{D}_{q,k} \right] \quad (7)$$

and its inverse is BCRLB matrix, i.e.,

$$\mathbf{B}_{\text{CRLB}}(\mathbf{x}_k) = J_m^{-1}(\mathbf{x}_k) \quad (8)$$

In (7), the term Σ_{ζ_k} represents the covariance matrix of the predictive state error, and its inverse matrix is the Fisher information matrix (FIM) of prior information. The term $\mathbb{E}_{\zeta_k^q} \left[\mathbf{D}_{q,k}^T \Sigma_{\mathbf{y}_k^q}^{-1} \mathbf{D}_{q,k} \right]$ is the FIM of the data. $\mathbf{D}_{q,k}$ denotes the Jacobian matrix of the measurement $\mathbf{y}_k^q$ with respect to predictive target's state ζ_k^q . The term $\Sigma_{\mathbf{y}_k^q}^{-1}$ represents the covariance matrix of the measurement noise [24,25].

It is known that FIM of the prior information only depends on the motion model of the target, while the FIM of the data is directly proportional to the transmitted power of the radar. Assume that the transmitted power is $\mathbf{P}_k = [P_{1,k}, P_{2,k}, \dots, P_{Q,k}]^T$ at time step k . When $P_{q,k}$ is zero, that is, there is no beam allocated to the q -th target for illumination at time step k , and the covariance matrix $\Sigma_{\mathbf{y}_k^q}^{-1}$ of the measurement error is a zero matrix. In order to describe the beam allocation more clearly, a group of binary variables $\mathbf{z}_k = [z_{1,k}, z_{2,k}, \dots, z_{Q,k}]^T$ is introduced here, where

$$z_{q,k} = \begin{cases} 1, & \text{if } P_{q,k} > 0 \\ 0, & \text{if } P_{q,k} = 0 \end{cases} \quad (9)$$

where $z_{q,k} = 1$ indicates that the radar illuminates target q at the k -th tracking time instant, and $z_{q,k} = 0$ is the opposite. Then, substitute (9) into (7), and we obtain [26,27]

$$J_m(\zeta_k^q) = \Sigma_{\zeta_k^q}^{-1} + z_{q,k} \mathbb{E}_{\zeta_k^q} \left[\mathbf{D}_{q,k}^T \Sigma_{\mathbf{y}_k^q}^{-1} \mathbf{D}_{q,k} \right] \quad (10)$$

whose inverse matrix, BCRLB matrix, can be used as an evaluation function of transmitted power $P_{q,k}$ and beam selection parameter $z_{q,k}$. The expression is as follows:

$$\mathbf{B}_{\text{CRLB}}(P_{q,k}, z_{q,k}) \Big|_{\zeta_k^q} = \left(\Sigma_{\zeta_k^q}^{-1} + z_{q,k} P_{q,k} \mathbb{E}_{\zeta_k^q} \left[\mathbf{D}_{q,k}^T \hat{\Sigma}_{\mathbf{y}_k^q}^{-1} \mathbf{D}_{q,k} \right] \right)^{-1} \quad (11)$$

in which the original covariance matrix of the measurement error $\Sigma_{\mathbf{y}_k^q}^{-1}$ is rewritten as a multiplication of $P_{q,k}$ and $\hat{\Sigma}_{\mathbf{y}_k^q}^{-1}$.

3.2. Problem Formulation of JBSPA

In JBSPA, since the target state ζ_k^q is unknown at the $k-1$ -th time step, a predictive state $\zeta_{k|k-1}^q$ is taken to calculate the FIM of data. In addition, in order to ensure the requirement of real-time in MTT, we use one-sampling measurement to calculate the FIM of data instead of the expectation of FIM of data. The final BCRLB matrix is given by [14]

$$\mathbf{B}_{\text{CRLB}}(P_{q,k}, z_{q,k}) \Big|_{\zeta_{k|k-1}^q} = \left(\Sigma_{\zeta_{k|k-1}^q}^{-1} + z_{q,k} P_{q,k} \hat{\mathbf{D}}_{q,k}^T \hat{\Sigma}_{\mathbf{y}_k^q}^{-1} \hat{\mathbf{D}}_{q,k} \Big|_{\zeta_{k|k-1}^q} \right)^{-1} \quad (12)$$

where $\hat{\mathbf{D}}_{q,k}$ denotes the Jacobian matrix of the measurement $\hat{\mathbf{y}}_k^q$ with respect to predictive target's state $\zeta_{k|k-1}^q$. $\hat{\Sigma}_{\mathbf{y}_k^q}$ is the approximation of the covariance matrix of measurement at the predictive state $\zeta_{k|k-1}^q$. In theory, the diagonal elements of the derived BCRLB matrix represent the corresponding lower bounds of the estimation MSEs of the target states, and thus, the tracking BCRLB of the target is defined as the trace of the BCRLB matrix, i.e.,

$$\Phi(P_{q,k}, z_{q,k}) = \text{Tr} \left(\mathbf{B}_{\text{CRLB}}(P_{q,k}, z_{q,k}) \Big|_{\zeta_{k|k-1}^q} \right) \quad (13)$$

where $\text{Tr}(\cdot)$ represents trace operator. In order to enhance the MTT performance as much as possible, and ensure the tracking performance of each target, we take the worst-case tracking BCRLB among targets as an evaluation of the MTT performance to optimize the resource allocation result, which is expressed as

$$F(\mathbf{P}_k, \mathbf{z}_k) = \max_q \sqrt{\Phi(P_{q,k}, z_{q,k})} \quad (14)$$

The problem of JBSPA is then described as follows: restricted by the predetermined maximum beam number M and the power budget P_{total} , the power allocation and the beam selection are jointly optimized for the purpose of maximizing the cumulative discounted worst-case BCRLB of MTT. The cost function is written as

$$\begin{aligned} \min_{\mathbf{P}_k, \mathbf{z}_k} \quad & \sum_{u=k}^K \gamma^{u-k} F(\mathbf{P}_k, \mathbf{z}_k) \\ \text{s.t.} \quad & \sum_{q=1}^Q z_{q,u} P_{q,u} = P_{total} \\ & \delta_k = \sum_{q=1}^Q z_{q,u} \leq M, \quad z_{q,u} \in \{0, 1\}, \\ & \begin{cases} P_{q,u} = 0 & z_{q,u} = 0 \\ \bar{P}_{q \min} \leq P_{q,u} \leq \bar{P}_{q \max} & z_{q,u} = 1 \end{cases} \quad \forall u = k, \dots, K, \forall q = 1, \dots, Q \end{aligned} \quad (15)$$

where constraints 1~3 are the total power constraint, the beam number constraint and the single beam power constraint, respectively. $\bar{P}_{q \min}$ and $\bar{P}_{q \max}$ are the minimum and the maximum of the single beam transmitted power, respectively. Due to the existence of binary beam selection parameters, the cost function (15) is a complex mixed-integer nonlinear optimization problem. As the discount factor is $\gamma = 0$, the cyclic minimizing method [28] and projection gradient algorithm can be used to solve the problem [14]. However, since this method needs to traverse all possible beam allocation solutions to find the optimal solution, the solving process is time-consuming. With this method, if there are a large number of targets to track, the real-time performance of MTT cannot be guaranteed. On the other hand, when the discount factor γ is non-zero, to solve the problem requires accurate future multi-step predictions of target state and BCRLB which is, however, hardly realized in real applications, especially in the scenarios involving maneuvering targets because the potential model-mismatch problem would amplify the prediction error and reduce the reliability of BCRLB seriously along with the tracking time increasing.

To address above issues, we discuss the JBSPA from MDP perspective, and solve it with a data-driven method: DRL. Compared with the optimization-based method, this method is able to avoid the multi-step prediction problem and complex solving process, and ensure the real-time performance of resource allocation. Besides that, in order to make the method suitable for maneuvering target scenarios, RNN with LSTM-GMM unit is used to learn the state transition regularity of maneuvering targets, and then realize the information prediction of the maneuvering target.

4. Joint Beam Selection and Power Allocation Strategy Based on Deep Reinforcement Learning

In this section, the original resource problem is first modeled as a model-free MDP. It is a complicated continuous controlling problem over action and state spaces. Therefore, the value-function-based DRLs, such as the DQN, are not applicable. For this problem, we choose to directly derive the Q function with respect to the policy parameters to calculate the policy gradient, and seek the optimal resource allocation policy along the descending direction of the policy gradient. In this paper, the proposed DRL-based method is named *DRL-JBSPA (DRL for Joint Beam Selection and Power Allocation)*.

4.1. Overall

The workflow of the DRL-JBSPA system is shown in Figure 1. The implementation at each tracking interval, the current measurement of targets (i.e., radial range, angle, Doppler and RCS) will be imported into the DRL system to perform the tasks of the information prediction and the resource allocation. The prediction network has also been incorporated into the DRL system to provide the DRL agent state at each decision step. The key idea behind this design is to realize the best beam and power control by learning from the massive interaction data between the DRL agent and the environment. It is noticed that this environment here is not real, it is established artificially and consists of the agent state, action and reward. In this application, according to the objective and the resource limitations of JBSPA, the agent state, the action and the reward are designed in the following.

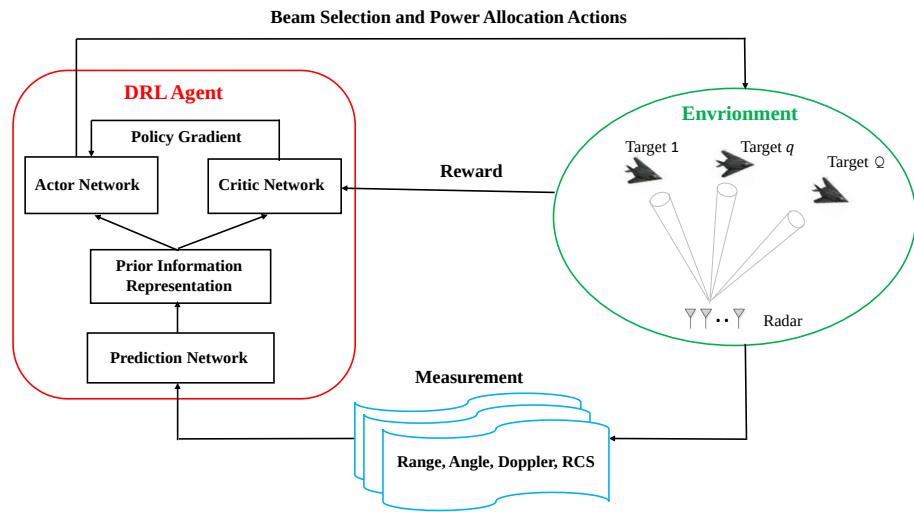


Figure 1. The system architecture of DRL-JBSPA.

STATE: The state of the DRL agent is defined as

$$\mathbf{s}_k = [\zeta_{k|k-1}^1, \dots, \zeta_{k|k-1}^q, \dots, \zeta_{k|k-1}^Q, \text{diag}(\Sigma_{\zeta_k^1}), \dots, \text{diag}(\Sigma_{\zeta_k^q}), \dots, \text{diag}(\Sigma_{\zeta_k^Q})]^T \quad (16)$$

where $\text{diag}(\cdot)$ represents the operation of taking the main diagonal elements. $\Sigma_{\zeta_k^q}$ denotes the covariance matrix of the one-step predictive state $\zeta_{k|k-1}^q$. According to Section 3.1, the covariance matrix represents the prediction error of target state, and the agent can determine how to allocate resource by comparing the prediction error of all targets. Moreover, the predictive states of all of the targets are also included in the agent state vector, whose aim is to utilize the target state information, such as the targets' ranges, the speeds and the RCS, to assist the agent in judgment. As $\gamma = 0$, the DRL agent tends to allocate more power and beam resources to the targets with larger prediction errors to ensure their tracking performances [14].

ACTION: Instead of using a binary variable $\mathbf{z}_k$ to describe the beam allocation, two action parameters δ_k and $\mathbf{X}_k$ are taken here to represent the generated beam number and determine which target to be illuminated in the next tracking time instant, respectively. δ_k is a scalar and $\mathbf{X}_k$ is

$$\mathbf{X}_k = [X_1, X_2, \dots, X_Q]^T, \quad X_q \in [0, 1], q = 1, 2, \dots, Q \quad (17)$$

Then, synthesizing the power allocation action and the two beam allocation actions into an action vector, we have

$$\mathbf{a}_k = [\delta_k, \mathbf{X}_k^T, \mathbf{P}_k^T]^T \quad (18)$$

REWARD: An intuitive reward function will benefit the learning efficiency of DRL. According to the objective function in (15), we set the reward function to the opposite of the worst-case BCRLB at the k -th time step, i.e.,

$$r_k(\mathbf{s}_k, \mathbf{a}_k) = -lF(\mathbf{P}_k, \mathbf{z}_k) \quad (19)$$

where l is a constant given artificially, and used for controlling the reward scale.

4.2. LSTM-GMM Based Prediction Network

The structure of prediction network is illustrated in Figure 2, where the LSTM is used with GMM output layer, say, LSTM-GMM. At the k -th time step, this network outputs the prior information representation $\mathbf{V}_k^q$ for target q with the measurement $\mathbf{y}_{k-1}^q$ as the input, which can be expressed in a mathematical form, i.e.,

$$\mathbf{V}_k^q = f_{\theta^R}(\mathbf{y}_{k-1}^q, h_{k-1}^q; \theta^R) \quad (20)$$

where $f_{\theta^R}(\cdot)$ represents an abstracted function of the RNN-GMM and θ^R is its parameter vector. Based on maximum likelihood, the loss function is

$$L(\theta^R) = -\ln(p(\xi_k^q)) \quad (21)$$

in which $p(\xi_k^q)$ is the probability density function (PDF) of the prior distribution of the extended target state. Since the state vector and the target RCS component are independent of each other, the original PDF $p(\xi_k^q)$ can be obtained by multiplying two independent PDFs, i.e.,

$$p(\xi_k^q) = p(\mathbf{x}_k^q)p(\psi_k^q) \quad (22)$$

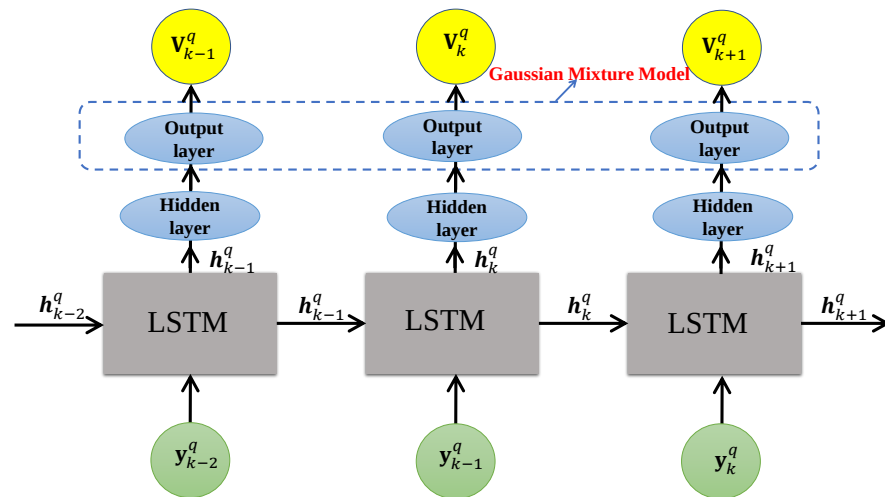


Figure 2. The prediction network.

In this section, in order to describe the state transition of the maneuvering target and the randomness and uncertainty of the target RCS, we assume that the distributions of the target state and the target RCS are two different Gaussian mixture models that are composed of K_t and K_r Gaussian distributions, respectively. The PDFs are expressed as

$$p(\mathbf{x}_k^q) = \sum_{k_t=0}^{K_t-1} \Pi_{q,k}^{k_t} \frac{\exp\left(-\frac{1}{2}\left(\mathbf{x}_k^q - \{\mu_{\mathbf{x}_k}^q\}^{k_t}\right)^T \Sigma_{\mathbf{x}_k^q}^{-1}\left(\mathbf{x}_k^q - \{\mu_{\mathbf{x}_k}^q\}^{k_t}\right)\right)}{(2\pi)^2 \left|\{\Sigma_{\mathbf{x}_k^q}\}^{k_t}\right|^{1/2}}, \quad \sum_{k_t=0}^{K_t-1} \Pi_{q,k}^{k_t} = 1 \quad (23)$$

$$p(\psi_k^q) = \sum_{k_r=0}^{K_r-1} \tilde{\Pi}_{q,k}^{k_r} \frac{\exp\left(-\frac{1}{2}\left(\psi_k^q - \{\mu_{\psi_k}^q\}^{k_r}\right)^T \Sigma_{\psi_k^q}^{-1}\left(\psi_k^q - \{\mu_{\psi_k}^q\}^{k_r}\right)\right)}{(2\pi)^2 \left|\{\Sigma_{\psi_k^q}\}^{k_r}\right|^{1/2}}, \quad \sum_{k_r=0}^{K_r-1} \tilde{\Pi}_{q,k}^{k_r} = 1 \quad (24)$$

where $\{\mu_{\mathbf{x}_k}^q\}^{k_t}$ and $\{\Sigma_{\mathbf{x}_k^q}\}^{k_t}$ are the mean and the covariance matrix of the k_t -th Gaussian distribution of target state $\mathbf{x}_k^q$, respectively. $\{\mu_{\psi_k}^q\}^{k_r}$ and $\{\Sigma_{\psi_k^q}\}^{k_r}$ denote the mean and the covariance matrix of the k_r -th Gaussian distribution of RCS ψ_k^q , respectively. $\Pi_{q,k}^{k_t}$ and $\tilde{\Pi}_{q,k}^{k_r}$ are the corresponding mixed coefficients. Since the target state has four dimensions, the mean of the covariance matrix of K_t -th mixed Gaussian distribution can be expressed as follows.

$$\begin{aligned} \{\mu_{\mathbf{x}_k}^q\}^{k_t} &= [\mu_{x_{q,k}}^{k_t}, \mu_{\dot{x}_{q,k}}^{k_t}, \mu_{y_{q,k}}^{k_t}, \mu_{\dot{y}_{q,k}}^{k_t}]^T \\ \{\Sigma_{\mathbf{x}_k^q}\}^{k_t} &= \begin{bmatrix} (\sigma_{x_k^q}^{k_t})^2 & -\rho_1^{k_t} \sigma_{x_k^q}^{k_t} \sigma_{\dot{x}_k^q}^{k_t} & 0 & 0 \\ -\rho_1^{k_t} \sigma_{x_k^q}^{k_t} \sigma_{\dot{x}_k^q}^{k_t} & (\sigma_{\dot{x}_k^q}^{k_t})^2 & 0 & 0 \\ 0 & 0 & (\sigma_{y_k^q}^{k_t})^2 & -\rho_2^{k_t} \sigma_{y_k^q}^{k_t} \sigma_{\dot{y}_k^q}^{k_t} \\ 0 & 0 & -\rho_2^{k_t} \sigma_{y_k^q}^{k_t} \sigma_{\dot{y}_k^q}^{k_t} & (\sigma_{\dot{y}_k^q}^{k_t})^2 \end{bmatrix} \end{aligned} \quad (25)$$

In (25), it is assumed that the target state in x -direction and y -direction are independent from each other. The terms $[\mu_{x_{q,k}}^{k_t}, \mu_{\dot{x}_{q,k}}^{k_t}]$ and $[\sigma_{x_k^q}^{k_t}, \sigma_{\dot{x}_k^q}^{k_t}]$ are the mean and the standard deviation of the K_r -th Gaussian distribution of the target state on x -direction, respectively, and $[\mu_{y_{q,k}}^{k_t}, \mu_{\dot{y}_{q,k}}^{k_t}]$ and $[\sigma_{y_k^q}^{k_t}, \sigma_{\dot{y}_k^q}^{k_t}]$ denote that on y -direction. $\rho_1^{k_t}$ and $\rho_2^{k_t}$ are the correlation coefficients of the location and the speed on x -direction and y -direction, respectively. Since there are two channels of RCS, the PDF $p(\psi_k^q)$ is a two-dimensional Gaussian, whose mean and covariance matrix are

$$\begin{aligned} \{\mu_{\psi_k}^q\}^{k_r} &= [\mu_{h_{q,k}^R}^{k_r}, \mu_{h_{q,k}^I}^{k_r}]^T \\ \{\Sigma_{\psi_k^q}\}^{k_r} &= \begin{bmatrix} (\sigma_{h_{q,k}^R}^{k_r})^2 & 0 \\ 0 & (\sigma_{h_{q,k}^I}^{k_r})^2 \end{bmatrix} \end{aligned} \quad (26)$$

where $\left(\mu_{h_{q,Tk}^{k_r}}, \mu_{h_{q,Tk}^{k_r}}\right)$ and $\left(\sigma_{h_{q,Tk}^{k_r}}, \sigma_{h_{q,Tk}^{k_r}}\right)$ represent the mean and the standard deviation of the K_r Gaussian distribution on channel R and channel I, respectively.

Based on the above derivations and analysis, the output vector of RNN at the k -th time step is given in (27), and its dimension is $11K_t + 5K_r$.

$$\mathbf{V}_k^q = \left[\mu_{\zeta_k^q}^T, \sigma_{\zeta_k^q}^T, \Pi_{\zeta_k^q}^T, \rho_{\mathbf{x}_k^q}^T \right]^T \quad (27)$$

where

$$\begin{cases} \mu_{\zeta_k^q} = \left[\mu_{x_{q,k}}^1, \dots, \mu_{x_{q,k}}^{K_t}, \mu_{\dot{x}_{q,k}}^1, \dots, \mu_{\dot{x}_{q,k}}^{K_t}, \mu_{y_{q,k}}^1, \dots, \mu_{y_{q,k}}^{K_t}, \mu_{\dot{y}_{q,k}}^1, \dots, \mu_{\dot{y}_{q,k}}^{K_t}, \mu_{h_{q,k}^R}^1, \dots, \mu_{h_{q,k}^R}^{K_r}, \mu_{h_{q,k}^I}^1, \dots, \mu_{h_{q,k}^I}^{K_r} \right]^T \\ \sigma_{\zeta_k^q} = \left[\sigma_{x_{q,k}}^1, \dots, \sigma_{x_{q,k}}^{K_t}, \sigma_{\dot{x}_{q,k}}^1, \dots, \sigma_{\dot{x}_{q,k}}^{K_t}, \sigma_{y_{q,k}}^1, \dots, \sigma_{y_{q,k}}^{K_t}, \sigma_{\dot{y}_{q,k}}^1, \dots, \sigma_{\dot{y}_{q,k}}^{K_t}, \sigma_{h_{q,k}^R}^1, \dots, \sigma_{h_{q,k}^R}^{K_r}, \sigma_{h_{q,k}^I}^1, \dots, \sigma_{h_{q,k}^I}^{K_r} \right]^T \\ \Pi_{\zeta_k^q} = \left[\Pi_{q,k}^1, \dots, \Pi_{q,k}^{K_t}, \tilde{\Pi}_{q,k}^1, \dots, \tilde{\Pi}_{q,k}^{K_t} \right]^T \\ \rho_{\mathbf{x}_k^q} = \left[\rho_1^1, \dots, \rho_1^{K_t}, \rho_2^1, \dots, \rho_2^{K_t} \right]^T \end{cases} \quad (28)$$

In the implementation, at each tracking interval, we will randomly select one from the mixed Gaussian distribution outputs as the prior distribution of the extended target state according to the mixing coefficients. Assuming the k_t -th Gaussian distribution of target state and the k_r -th Gaussian distribution of RCS is selected, then the predictive extended target state and the corresponding FIM of prior information $J_P(\zeta_k^q)$ are given by

$$\zeta_{k|k-1}^q = \left[\mu_{x_{q,k}}^{k_t}, \mu_{\dot{x}_{q,k}}^{k_t}, \mu_{y_{q,k}}^{k_t}, \mu_{\dot{y}_{q,k}}^{k_t}, \mu_{h_{q,k}^R}^{k_r}, \mu_{h_{q,k}^I}^{k_r} \right]^T \quad (29)$$

$$J_P(\zeta_k^q) = \Sigma_{\zeta_k^q}^{-1} = \left[\text{diagblk} \left\{ \left\{ \Sigma_{\mathbf{x}_k^q} \right\}^{k_t}, \left\{ \Sigma_{\psi_k^q} \right\}^{k_r} \right\} \right]^{-1} \quad (30)$$

where $\text{diagblk}(\cdot)$ refers to the operation of the diagonal block matrix. Taking (29) and (30) into (13), we obtain the predicted tracking BCRLB of the q -th target at the k -th time step.

In the training stage, if the training set $\mathbf{D} = \left\{ (\tilde{\mathbf{y}}_1^m, \tilde{\zeta}_1^m), (\tilde{\mathbf{y}}_2^m, \tilde{\zeta}_2^m), \dots, (\tilde{\mathbf{y}}_{L_{\tilde{\zeta}}}^m, \tilde{\zeta}_{L_{\tilde{\zeta}}}^m) \right\}_{m=1}^M$ consists of M tracks with length $L_{\tilde{\zeta}}$, the loss of RNN is calculated by

$$L(\theta^R) = \frac{1}{2} \sum_{m=1}^M \sum_{k=1}^{L_{\tilde{\zeta}}} \left(\zeta_k^m - \mu_{\zeta_k}^m \right)^T \left(\Sigma_{\zeta_k}^m \right)^{-1} \left(\zeta_k^m - \mu_{\zeta_k}^m \right) \quad (31)$$

The time step of LSTM is set to $L_{\tilde{\zeta}}$ and the Adam optimizer [29] is utilized with dropout. In the implementation, the probability of switching off the neuron output in the dropout function is set to 0.3.

As the prediction network structure shown in Figure 2, the structure of this network is established on a single-layer LSTM unit whose gate functions and nonlinear layer are all composed of 64 neurons. At the k -th time step, the observation vector is first normalized and then taken into the LSTM unit to execute “remember” and “forget” operations, and we obtain the final output $\mathbf{O}_k$ after the data flow through the hidden layer and the output layer, where $\mathbf{O}_k^{1 \rightarrow 4K_t + 2K_r}$ is the extended target state component, $\mathbf{O}_k^{4K_t + 2K_r + 1 \rightarrow 8K_t + 4K_r}$ is the standard deviation of each component of the extended target state, $\mathbf{O}_k^{8K_t + 4K_r + 1 \rightarrow 9K_t + 5K_r}$ are the mixed coefficients, and $\mathbf{O}_k^{9K_t + 5K_r + 1 \rightarrow 11K_t + 5K_r}$ are the correlation coefficients.

Since the standard deviation and the mixing coefficient of each target state component are strictly non-negative, the elements $\mathbf{O}_k^{4K_t+2K_r+1 \rightarrow 8K_t+4K_r}$ and $\mathbf{O}_k^{8K_t+4K_r+1 \rightarrow 9K_t+5K_r}$ should be operated as follows.

$$\hat{\mathbf{O}}_k^{4K_t+2K_r+1 \rightarrow 8K_t+4K_r} = \exp(\mathbf{O}_k^{4K_t+2K_r+1 \rightarrow 8K_t+4K_r}) \quad (32)$$

$$\hat{\mathbf{O}}_k^{8K_t+4K_r+1 \rightarrow 9K_t+5K_r} = \exp(\mathbf{O}_k^{8K_t+4K_r+1 \rightarrow 9K_t+5K_r}) \quad (33)$$

Additionally, considering that the output $\mathbf{O}_k$ is a standardized result, the final estimated target state is calculated using the mean and the standard deviation of the samples, i.e.,

$$\begin{cases} \mu_{\zeta_k^q} = \mathbf{O}_k^{1 \rightarrow 4K_t+2K_r} \odot \bar{\mathbf{s}}_{\zeta} + \bar{\zeta} \\ \sigma_{\zeta_k^q} = \left(\hat{\mathbf{O}}_k^{4K_t+2K_r+1 \rightarrow 8K_t+4K_r} \odot \bar{\mathbf{s}}_{\zeta} \right)^2 \\ \Pi_{\zeta_k^q} = \hat{\mathbf{O}}_k^{8K_t+4K_r+1 \rightarrow 9K_t+5K_r} / \sum \hat{\mathbf{O}}_k^{8K_t+4K_r+1 \rightarrow 9K_t+5K_r} \\ \rho_{\mathbf{x}_k^q} = \mathbf{O}_k^{9K_t+5K_r+1 \rightarrow 11K_t+5K_r} \end{cases} \quad (34)$$

where $\bar{\mathbf{s}}_{\zeta}$ and $\bar{\zeta}$ are the mean and standard deviation of the samples, and the calculation is as follows.

$$\begin{cases} \bar{\zeta} = \frac{1}{ML_{\zeta}} \sum_{m=1}^M \sum_{l=1}^{L_{\zeta}} \zeta_l^m \\ \bar{\mathbf{s}}_{\zeta} = \sqrt{\frac{1}{ML_{\zeta}-1} \sum_{m=1}^M \sum_{l=1}^{L_{\zeta}} (\zeta_l^m - \bar{\zeta})^2} \end{cases} \quad (35)$$

4.3. Actor-Critic-Based DRL for JBSPA

Considering that the MDP of JBSPA is a continuous controlling problem and the traditional Q-table method is not available, we adopt the PG-based method to solve it. By introducing the policy function $\pi_{\theta}(\mathbf{a}|\mathbf{s})$, the original problem (5) can be characterized as a classical MDP problem, which aims to find the optimal policy parameter θ^* that makes the Q-value function achieve the maximum, i.e.,

$$\theta^* = \arg \max_{\theta} Q^{\pi_{\theta}}(\mathbf{s}_k, \mathbf{a}_k) \quad (36)$$

where the Q function is calculated by

$$Q^{\pi_{\theta}}(\mathbf{s}_k, \mathbf{a}_k) = \mathbb{E}_{\mathbf{s}_{k+1} \sim \rho^{\pi}, \mathbf{a}_k \sim \pi_{\theta}(\mathbf{a}_k|\mathbf{s}_k)} \left[r(\mathbf{s}_k, \mathbf{a}_k) + \gamma \mathbb{E}_{\mathbf{a}_{k+1} \sim \pi_{\theta}(\mathbf{a}_{k+1}|\mathbf{s}_{k+1})} [Q^{\pi_{\theta}}(\mathbf{s}_{k+1}, \mathbf{a}_{k+1})] \right] \quad (37)$$

in which the symbol ρ^{π} refers to the state space with policy $\pi_{\theta}(\mathbf{a}|\mathbf{s})$. According to the definition of DRL state for our problem, we know that the DRL state is determined by the current target measurement information. However, the target measurement is always random duo to the existence of observation noise. Therefore, the DRL state transition is not deterministic, i.e.,

$$p(\mathbf{s}_{k+1}|\mathbf{s}_k, \mathbf{a}_k) \neq 1 \quad (38)$$

According to (37), if the random policy is used at each decision time step, a large number of samples are required to calculate the Q value of all possible actions in the continuous action space or high-dimensional discrete action space. In order to reduce the computational complexity, the deterministic policy is adopted in this section, which is expressed as

$$\mathbf{a}_k = \mu_{\theta}(\mathbf{s}_k) \quad (39)$$

where $\mu_{\theta}(\cdot)$ is the deterministic policy function. Then, take (39) into (37), simplify, and have

$$Q^{\mu_{\theta}}(\mathbf{s}_k, \mathbf{a}_k) = \mathbb{E}_{\mathbf{s}_{k+1} \sim \rho^{\pi}} [r(\mathbf{s}_k, \mathbf{a}_k) + \gamma Q^{\pi_{\theta}}(\mathbf{s}_{k+1}, \mathbf{a}_{k+1})] \quad (40)$$

Next, derive the Q-value function with respect to ϑ , and obtain the following deterministic policy gradient:

$$\nabla_{\vartheta} \mathcal{J}^{\pi}(\vartheta) = \mathbb{E}_{\mathbf{s}_{k+1} \sim \rho^{\pi}} \left[\nabla_{\vartheta} \mu_{\vartheta}(\mathbf{s}_{k+1}) \nabla_{\mathbf{a}} Q^{\mu_{\vartheta}}(\mathbf{s}_{k+1}, \mathbf{a}_{k+1}) \Big|_{\mathbf{a}_{k+1} = \mu_{\vartheta}(\mathbf{s}_{k+1})} \right] \quad (41)$$

Subsequently, the equivalence between original problem and the established MDP problem is discussed in this section.

$$\begin{aligned} \max_{\vartheta} Q^{\mu_{\vartheta}}(\mathbf{s}_k, \mathbf{a}_k) &= \max_{\vartheta} [\mathbb{E}_{\mathbf{s}_{k+1} \sim \rho^{\pi}} [r(\mathbf{s}_k, \mathbf{a}_k) + \gamma Q^{\pi_{\vartheta}}(\mathbf{s}_{k+1}, \mathbf{a}_{k+1})]] \\ &= \min_{P_{q,k}, q=1, \dots, Q} \left[\Phi(P_{q,k}) + \gamma \mathbb{E} \left[\sum_{m=k+1}^K \gamma^{m-k-1} \Phi(P_{q,m}) \right] \right] \\ &= \min_{P_{q,k}, q=1, \dots, Q} \mathbb{E} \left[\sum_{m=k+1}^K \gamma^{m-k} \Phi(P_{q,m}) \right] \end{aligned} \quad (42)$$

It can be seen from (42) that the MDP problem (36) is essentially the expectation form of original problem (15). In other words, the aim of the optimization problem is transformed to maximizing the expectation of the cumulative discounted MTT performance. The usage of the reinforcement learning makes it feasible to perform power allocation considering the long-term and short-term tracking performance by estimating the Q value to approximate the cumulative discounted MTT BCRLB.

Based on the derived deterministic policy gradient, this paper uses the deep deterministic policy gradient (DDPG) [22] algorithm to learn the optimal power allocation strategy based on the actor-critic framework. DDPG is a reinforcement learning algorithm that combines deep learning and the deterministic policy gradient. Its biggest advantage is that it can learn more effectively in the continuous action space. The structure of the DDPG mainly consists of two deep networks, i.e., the actor network and the critical network.

In the calculation of the Q value, the DDPG draws lessons from the idea of deep Q learning and adopts the target network with exactly the same structure as the critic-actor network to ensure the stability of network training. The DDPG prefers to update the target network parameters slightly by “soft” updating the target network, and achieve the improvement of learning stability. This operation is similar to the regular replication of parameters to the target network in the DQN. The soft update of the target network parameters is expressed as follows:

$$\begin{aligned} \theta^{\mu'} &:= \tau \theta^{\mu} + (1 - \tau) \theta^{\mu'} \\ \theta^{Q'} &:= \tau \theta^Q + (1 - \tau) \theta^{Q'} \end{aligned} \quad (43)$$

where θ^{μ} and θ^Q are the parameters of the actor network $\mu(\cdot | \theta^{\mu})$ and the critic network $Q(\cdot | \theta^Q)$, respectively. $\theta^{\mu'}$ and $\theta^{Q'}$ are the parameters of the target networks $\mu'(\cdot | \theta^{\mu'})$ and $Q'(\cdot | \theta^{Q'})$, respectively. τ is the soft updating parameter which takes always a small value.

Exploration behavior is also introduced to make the DRL agent more experienced, with which the DRL agent is able to try unexplored areas to discover the improvement opportunity. At the k -th decision time step, the performed action is given by

$$\mathbf{a}_k = \mu(\mathbf{s}_k | \theta^{\mu}) + \mathcal{O} \quad (44)$$

where $\mathcal{O}$ is the random noise for agent exploration.

In order to further ensure the stability of learning, the experience relay technique is used to eliminate the strong temporal autocorrelation inherent to DRL, and can reduce the variance of the estimation of Q-values. At each training step, assuming that H experience

samples are randomly selected from the experience replay buffer $\mathbf{B}$ to update the critic network and the actor network, the loss function of the critic network is written as

$$L^Q(\theta^Q) = \frac{1}{H} \sum_{j=1}^H \left(Q(\mathbf{s}_j, \mathbf{a}_j | \theta^Q) - Q_j^{\text{Tar}} \right)^2 \quad (45)$$

where Q_j^{Tar} denotes the target Q value, and is used as “labels” for the estimation of the Q-values. Q_j^{Tar} is calculated by

$$Q_j^{\text{Tar}} := r_j + \gamma Q'(\mathbf{s}_j, \mu'(\mathbf{s}_j | \theta^{\mu'}) | \theta^{Q'}) \quad (46)$$

According to (41), the policy gradient with one sampling can be obtained by

$$\nabla_{\theta^\mu} \mathcal{J}^\pi(\theta^\mu) \approx \frac{1}{H} \sum_{j=1}^H \left[\nabla_{\theta^\mu} \mu(\mathbf{s} | \theta^\mu) \Big|_{\mathbf{s}=\mathbf{s}_j} \nabla_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a} | \theta^Q) \Big|_{\mathbf{s}=\mathbf{s}_j, \mathbf{a}=\mu(\mathbf{s}_j | \theta^\mu)} \right] \quad (47)$$

The parameter of the actor network θ^μ is synchronously updated based on the chain rule of gradient propagation. More details of the DRL-JBSPA are given in Algorithm 1.

Algorithm 1: DRL-JBSPA.

- 1: Randomly initialize parameters θ^μ and θ^Q of actor network $\mu(\cdot | \theta^\mu)$ and critic network $Q(\cdot | \theta^Q)$;
 - 2: Initialize target networks $\theta^{\mu'} \leftarrow \theta^\mu, \theta^{Q'} \leftarrow \theta^Q$;
 - 3: Initialize replay buffer $\mathbf{B}$;
 - 4: **for** each episode **do**
 - 4: Initialize random process $\mathcal{O}$ for action exploration;
 - 5: Receive initial state $\mathbf{s}_1$ derived from the prediction network
 $\mathbf{s}_1 = f_{\theta^R}(\mathbf{y}_0^q, h_0^q)$;
 - 6: **for** each step k of episode **do**
 - 7: Select action $\mathbf{a}_k = \mu(\mathbf{s}_k | \theta^\mu) + \mathcal{O}$;
 - 8: Execute action $\mathbf{a}_k$ and observe state $\mathbf{s}_{k+1}$ and reward r_k
 $\mathbf{s}_{k+1} = f_{\theta^R}(\mathbf{y}_k^q(\mathbf{a}_k), h_k^q)$
 $r_k = -F(\mathbf{P}_k, \mathbf{z}_k)$;
 - 9: Store transition sample $(\mathbf{s}_k, \mathbf{a}_k, r_k, \mathbf{s}_{k+1})$ into replay buffer $\mathbf{B}$;
 - 10: Sample from $\mathbf{B}$ a minibatch of H transitions;
 - 11: Calculate the target Q value:
 $Q_j^{\text{Tar}} := r_j + \gamma Q'(\mathbf{s}_j, \mu'(\mathbf{s}_j | \theta^{\mu'}) | \theta^{Q'})$;
 - 12: Update critic network by minimizing the loss:
 $L^Q(\theta^Q) = \frac{1}{H} \sum_{j=1}^H \left(Q(\mathbf{s}_j, \mathbf{a}_j | \theta^Q) - Q_j^{\text{Tar}} \right)^2$;
 - 13: Update the actor network using the sampled policy gradients:
 $\nabla_{\theta^\mu} \mathcal{J}^\pi(\theta^\mu) \approx \frac{1}{H} \sum_{j=1}^H \left[\nabla_{\theta^\mu} \mu(\mathbf{s} | \theta^\mu) \Big|_{\mathbf{s}=\mathbf{s}_j} \nabla_{\mathbf{a}} Q(\mathbf{s}, \mathbf{a} | \theta^Q) \Big|_{\mathbf{s}=\mathbf{s}_j, \mathbf{a}=\mu(\mathbf{s}_j | \theta^\mu)} \right]$;
 - 14: Update the target networks:
 $\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$
 $\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$;
 - 15: **end for**
 - 16: **end for**
-

The used actor-critic network is shown in Figure 3, where the actor network contains two hidden layers and three output layers. The hidden layers are dense layers consisting of 256 neurons with ReLU activation, and the three output layers are all dense layers and

share the same hidden layers. For different output actions, three output layers are named Actor 1, Actor 2 and Actor 3, respectively. Actor 1 consists of M neurons, and Actor 2 and Actor 3 are both composed of Q neurons.

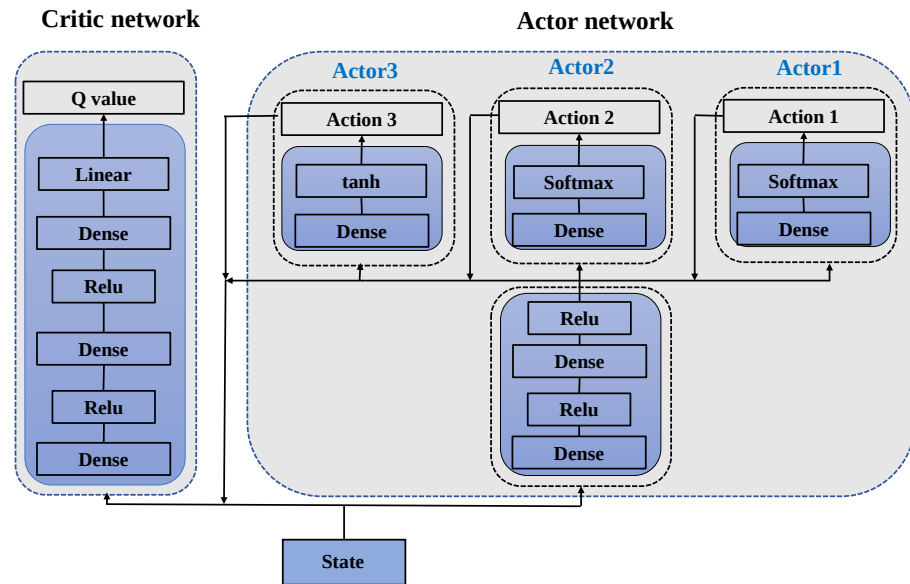


Figure 3. The Actor-Critic network.

At the k -th time step, the DRL state s_k is first input to the hidden layer in the actor network, and then, the output of the hidden layer will be taken into Actor 1. The activation function of Actor 1 is set as the softmax function $s_f(\cdot)$, which is used to derive the probability of all possible beam numbers:

$$\mathbf{a}^1 = s_f(\mathbf{w}_{A_1} \mathbf{h}_k^A + \mathbf{b}_{A_1}) \quad (48)$$

where $\mathbf{h}_k^A$ refers to the output of the hidden layer, $\mathbf{w}_{A_1}$ and $\mathbf{b}_{A_1}$ represent the weight and the bias of Actor 1, respectively. By selecting the superscript index corresponding to the maximum probability in $\mathbf{a}^1$, the number of beams generated by the radar at the k -th time step can be determined, i.e., $\delta_k = \arg \max \mathbf{a}^1$.

Next, concatenate $\mathbf{a}^1$ with the output of the hidden layer, and then send it to Actor 2. Actor 2 is responsible for determining which target will be illuminated at the next tracking time step. For this purpose, the softmax activation function is used here to output the probability of each target illuminated. The output of Actor 2 is calculated by

$$\mathbf{a}^2 = s_f(\mathbf{w}_{A_2} [\mathbf{h}_k^A, \mathbf{a}^1] + \mathbf{b}_{A_2}) \quad (49)$$

in which $\mathbf{w}_{A_2}$ and $\mathbf{b}_{A_2}$ are the weight and the bias of Actor 2. Then, the target to be illuminated can be determined by selecting the largest δ_k element from $\mathbf{a}^2$, which is $\mathbf{X}_k = \arg \max (\mathbf{a}^2, \delta_k)$.

In the same way, we concatenate the outputs of the hidden layer, actor 1 and actor 2, and import it into actor 3. Actor 3's activation function is set to the tanh function. The calculation formula is as follows:

$$\mathbf{a}^3 = \tanh(\mathbf{w}_{A_3} [\mathbf{h}_k^A, \mathbf{a}^1, \mathbf{a}^2] + \mathbf{b}_{A_3}) \quad (50)$$

where $\mathbf{w}_{A_3}$ and $\mathbf{b}_{A_3}$ are the weight and the bias of Actor 3. Select the elements in $\mathbf{a}^3$ with index $\mathbf{X}_k$ to form the vector $\mathbf{P}_k$, and then normalize it to obtain the ratio of each target's

allocated power to the total transmitted power. So far, by selection and splicing the outputs of the three output layers, we finally obtain the complete action vector $\mathbf{a}_k = [\delta_k, \mathbf{X}_k^T, \mathbf{P}_k^T]^T$.

The critic network takes the concatenated action vector $\tilde{\mathbf{a}}_k = [\mathbf{a}^1, \mathbf{a}^2, \mathbf{a}^3]$ and current agent state $\mathbf{s}_k$ as input and outputs the corresponding Q value. The critic network consists of two dense hidden layers and an output layer, where each of the hidden layers consists of 256 neurons with ReLU activation, and the output layer is a linear neuron. In our implementation, the Adam optimizer is used to train the actor-critic network, and the learning rate is set to $1e-4$. The soft updating parameter τ is set to $1e-3$. The deep learning library TensorFlow is taken to implement the networks on a personal computer with Intel Core i5-7500 CPU and 16GB RAM. NVIDIA GeForce GTX 1080TI GPU is used to accelerate the training process.

5. Simulation Results

In practice, the radar system has to face the challenge that the target number is larger than the beam number. Therefore, the simulation experiments are carried out under the condition that the maximum number of beams is less than the target number. Specifically, simulation 1 is conducted to investigate the proposed method by comparing the optimization-based method with the discounted factor $\gamma = 0$, simulation 2 is taken to discuss the effect of the discounted factor γ on the resource allocation results and the tracking performance.

The targets and the radar are depicted in the Cartesian coordinate system. Without loss of generality, the parameters of the radar system are set as follows: the radar system is assumed to be located at (0 km, 0 km), the signal effective bandwidth and effective time duration of the q th beam are set as $B_{q,k} = 2$ MHz and $T_{q,k} = 1$ ms, respectively. The carrier frequency of each beam is set as 1 GHz, and thus, the carrier wavelength is $\lambda_q = 0.3$ m. The time interval between successive frames is set as 2 s, and a sequence of 50 frames of data are utilized to support the simulation. The lower and upper bounds of the power of the q th beam are set to $\bar{P}_{q,\min} = 0.1P_{total}$ and $\bar{P}_{q,\max} = P_{total}$, respectively. The maximum number of beams is 5 and the target number is 8. For better illustration of the effectiveness of the proposed method, the initial positions and initial velocities of the targets are set the same, i.e., (1e5 m, 1e5 m) and (± 200 m/s, ± 200 m/s). The turning frequency of turn motion targets is set to 0.0035 Hz. The motion parameters of each target have been listed in Table 1, where the symbol “*” indicates that a motion mode does not include this motion parameter. The real moving trajectories of the three targets are depicted in Figure 4. As shown in Figure 4, Target 1 and Target 4 are doing right-turn motion, Target 2, Target 5 and Target 8 are doing left-turn motion; Target 3, Target 6 and Target 7 are doing uniform linear motion.

Table 1. Target motion parameters.

	Motion Mode	Turning Frequency	Initial Velocity
Target 1	Right-turn motion	0.0035	(−200 m/s, −200 m/s)
Target 2	Left-turn motion	0.0035	(−200 m/s, −200 m/s)
Target 3	Uniform linear motion	*	(−200 m/s, −200 m/s)
Target 4	Right-turn motion	0.0035	(−200 m/s, −200 m/s)
Target 5	Left-turn motion	0.0035	(200 m/s, −200 m/s)
Target 6	Uniform linear motion	*	(200 m/s, −200 m/s)
Target 7	Uniform linear motion	*	(200 m/s, 200 m/s)
Target 8	Left-turn motion	0.0035	(−200 m/s, 200 m/s)

The prediction network is first trained for deriving prior information representation, where the training examples are the measurements of the targets’ states with the trajectories of left-turn motion, right-turn motion, and uniform linear motion, and the corresponding labels are true targets’ states. The initial position and the speed of the targets are randomly generated within [50 km, 150 km] and [−300 m/s, 300 m/s] on x -axis and y -axis, respec-

tively. The turning frequencies are randomly generated within [0.001 Hz, 0.008 Hz]. In the process of generating training examples, the noise power added in measurements is determined by the allocated transmit power that is randomly generated within the interval $[\bar{P}_{q\min}, \bar{P}_{q\max}]$. We trained RNN over 200,000 epochs, and AC-JBSPA over 500,000 epochs (i.e., 500,000 transition samples) in an off-line manner.

Simulation 1: In this section, a series of simulation experiments are carried out to validate the effectiveness of the aforementioned methods, in which we adopt root mean square error (RMSE) to measure tracking performance, and it is given by

$$\text{RMSE}_k = \sqrt{\frac{1}{N_{MC}} \sum_{j_m=1}^{N_{MC}} \|\mathbf{x}_k^q - \hat{\mathbf{x}}_{k|k}^{q,j_m}\|_2^2} \quad (51)$$

where N_{MC} is the number of Monte Carlo trials, $\|\cdot\|_2$ denotes the 2-norm, and $\hat{\mathbf{x}}_{k|k}^{q,j_m}$ is the estimate of the q th target's state at the j_m -th trial with Kalman filter [30].

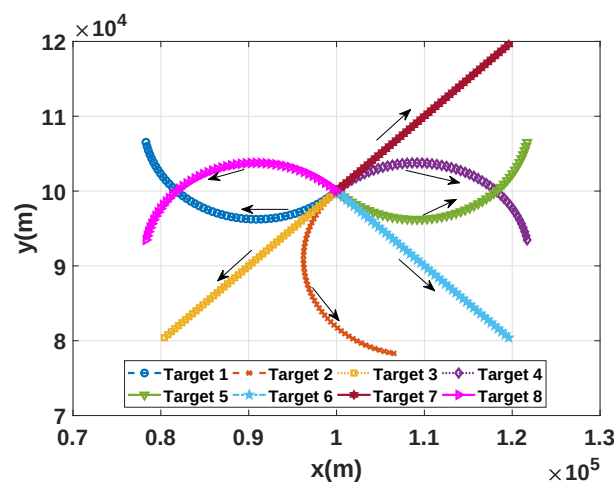


Figure 4. Real moving trajectories of targets.

We first compare the proposed method with the allocation methods introduced in [14] as the amplitude of target reflectivity is set to a constant 1. The optimization-based method is based on the optimization techniques, where the CV model is utilized for the prediction of target's state and the calculation of the FIM of prior information. Since the optimization-based method only focuses on the performance of the current time instant, the discounted factor is set to $\gamma = 0$ for a fair comparison. It is worth noting that the tracking RMSE of all the targets are set the same at the initial time, and in the figures about tracking performance, all of the RMSE curves start after the first resource allocation.

The comparison is shown in Figure 5, in which the curves labeled “ $P_{\text{Opt}}(\text{RMSE})$ ” and “ $P_{\text{DL}}(\text{RMSE})$ ” represent the tracking RMSEs averaged over 50 trails achieved with the optimization-based power allocation method, and proposed power allocation method, respectively. The curves labeled “ $P_{\text{Opt}}(\text{BCRLB})$ ” and “ $P_{\text{DL}}(\text{BCRLB})$ ” are the corresponding tracking BCRLBs. It can be seen that the tracking of the worst RMSEs are approaching the corresponding worst BCRLBs along with the number of measurements increasing. From the about 20th frame, the performance of the proposed method is about 10% higher than that of the optimization-based method with the CV model.

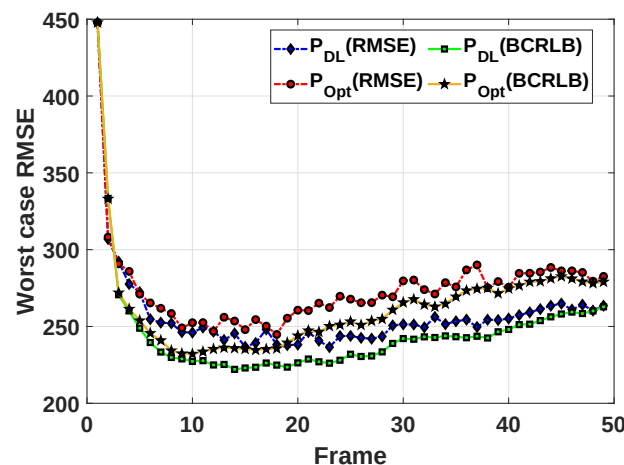


Figure 5. The worst case RMSEs achieved with different power allocation methods versus corresponding worst case BCRLBs.

The estimation RMSEs and the corresponding power allocation results are presented in Figures 6 and 7, respectively. In Figure 6a, the power allocation solution given by the proposed method makes the tracking RMSEs of different targets closer than that of the optimization-based method, which implies the resource utilization efficiency of the power allocation solution based on the learned policy (shown in Figure 7a) is higher than that of the optimization-based method with the CV model (shown in Figure 7b). In other words, there is room for the solution given by optimization-based method to continue to be optimized.

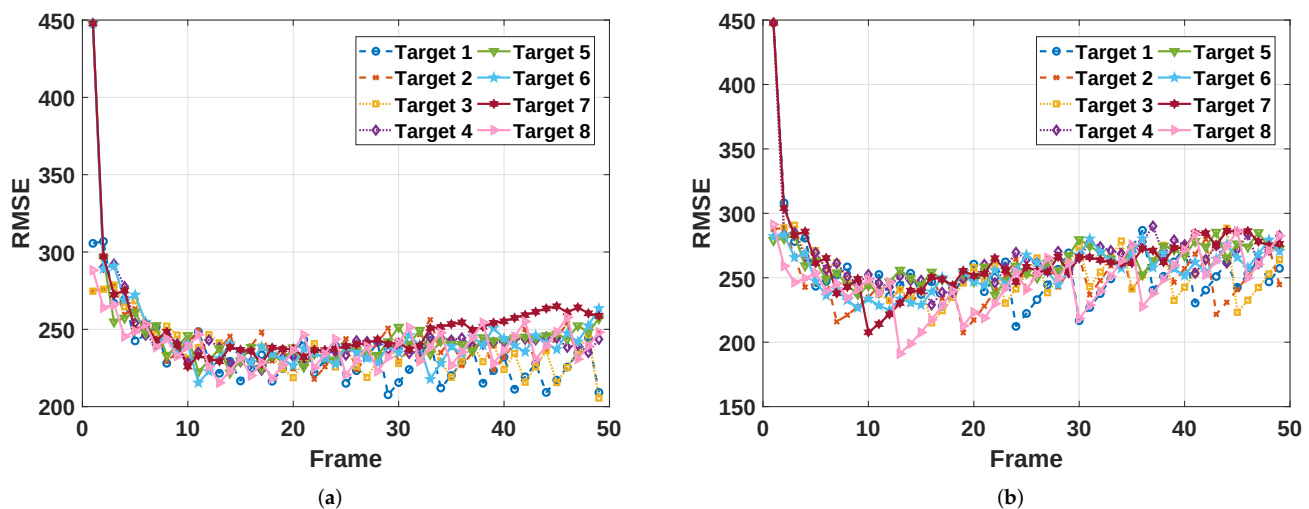


Figure 6. Tracking RMSEs of different targets. (a) Proposed power allocation method. (b) Optimization-based power allocation method.

In Figure 7a,b, it can be seen that more power and beam resource tend to be allocated to targets that are moving away from radar, i.e., target 4, target 5, target 6, and target 7, for achieving a larger gain of tracking performance. Actually, the resource allocation is not only determined by targets' radial distances but also by radius speeds, especially when the targets are close enough. As shown in Figure 7a, comparing the allocation results of these targets that are approaching the radar, e.g., Target 1, Target 2, Target 3, and Target 8, we discover that more power tends to be allocated to the beam pointing to the closer Target 1 because Target 1's larger radial speeds may cause a larger BCRLB. When using the CV model for prediction, the beam selection and the power allocation are mainly determined by

comparing the radius distances of different targets. For example, for the resource allocation solution of Target 1, Target 2, Target 3 shown in Figure 7b, more power and beam resources tend to be allocated to the farther targets, Target 1 and Target 2, from about frame 20. Therefore, we believe the proposed method can provide more accuracy prior information of targets, which benefits the utilization effectiveness of resource allocation.

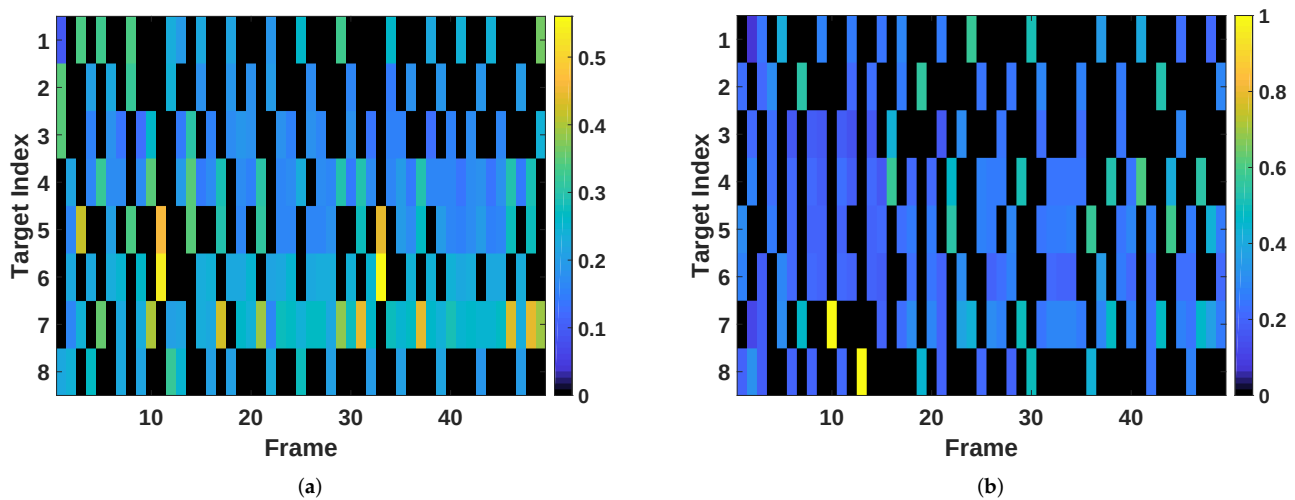


Figure 7. The simultaneous multibeam power allocation results. (a) Proposed power allocation method. (b) Optimization-based power allocation method.

To better illustrate the effectiveness of the RNN-based prediction network, an additional simulation experiment is conducted, where the optimization-based method is used to solve cost function (5) based on the prior information representation derived from the prediction network. In Figure 8, the curve labeled “ $P_{Opt}^{RNN}(RMSE)$ ” represents the worst case tracking RMSE of the RNN-optimization-based method. It can be seen that this curve almost coincides with that of the proposed data-driven method and is also higher than that of the optimization-based method. This shows that the cost function (5) can be optimally solved with the method in [14] once a one-step predictive state is accurate enough. According to the comparison shown in Figure 8, it is known that, as $\gamma = 0$, the tracking performance improvements brought from the proposed method are mainly due to the usage of the RNN-based prediction network.

Furthermore, a prediction performance comparison of the RNN-based prediction network and the Kalman filter is given in Figure 9, where the power is fixed during tracking. We first compare the RMSE of the predictive state of Target 2 (doing left turns) with the RNN and Kalman filter. The results are shown in Figure 9a, in which the prediction performance of RNN is higher than that of Kalman filter with the CV model because the model mismatch problem is avoided to a certain extent. Additionally, Figure 9b shows a comparison of the two methods of Target 3 (doing uniform linear motion). Here, Target 3 is taken in order to avoid the influence of model-mismatch. It can be seen that, different from that of Target 2, the RMSE curves of the two methods almost coincide, which implies that the performance of the RNN-based prediction network is very close (almost the same) to that of the Kalman filter with the true model (CV model), and thus, we believe that the RNN-based prediction network is able to replace the model-based prediction for target tracking in the applications of radar system resource allocation.

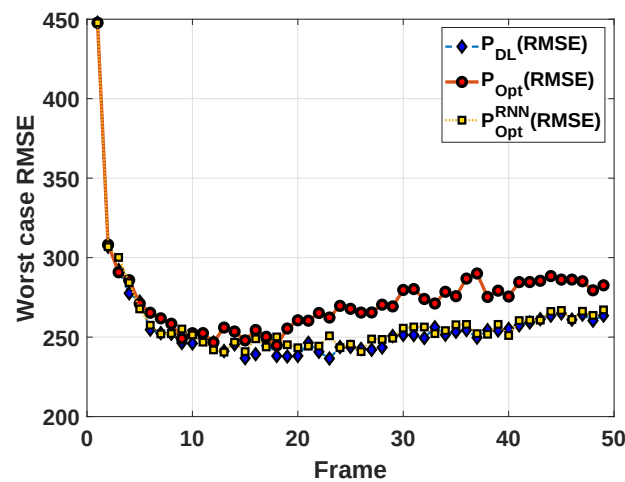


Figure 8. Worst case RMSEs achieved by the optimization-based method based on the prior representation derived by the prediction network.

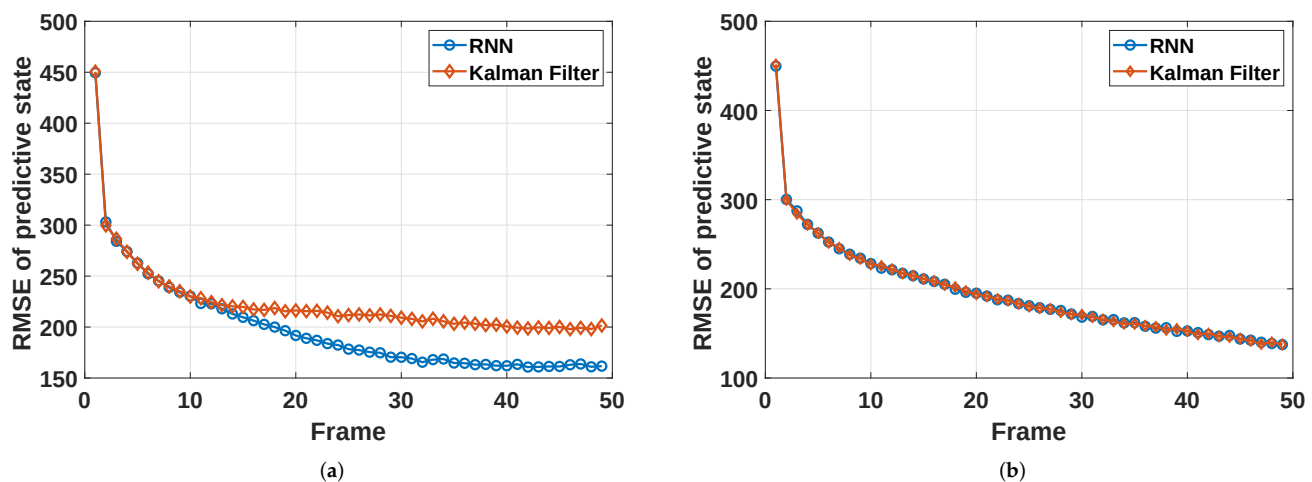


Figure 9. RMSE comparison of predictive state of different methods. (a) Target 2 (b) Target 1.

Next, we discuss the two methods with given RCS models. The amplitude sequences of RCS are given in Figure 10. In Figure 11, the curves labeled “ $P_{Opt}(RMSE)$ ” and “ $P_{DL}(RMSE)$ ” are the worst tracking RMSEs averaged over 50 trails achieved with the optimization-based method and the proposed method, respectively. Similar to that in the first case, the tracking performance of the proposed data-driven-based method is also superior to that of the optimization-based method. In Figures 12 and 13, we present the tracking RMSE and the corresponding resource allocation results of each target with the two methods. In Figure 12, the tracking RMSE curves of targets given by the proposed method is closer than that of the optimization-based method. This indicates that the resource utilization efficiency of the optimization-based method is lower than that of the proposed method. Comparing the resource allocation results of Target 1, Target 2 and Target 3 in Figure 13a, we observe that the revisit frequency of radar for Target 3 is higher than that of the other two. This phenomenon can be explained in that a high revisit frequency benefits the tracking performance improvement of the targets with a larger radial speed. On the other hand, more power and beam resource tend to be allocated to these targets with smaller RCS amplitude, i.e., Target 1, Target 2 and Target 3, which aim to improve SNR of echoes such that the worst-case tracking performance among multiple targets is maximized.

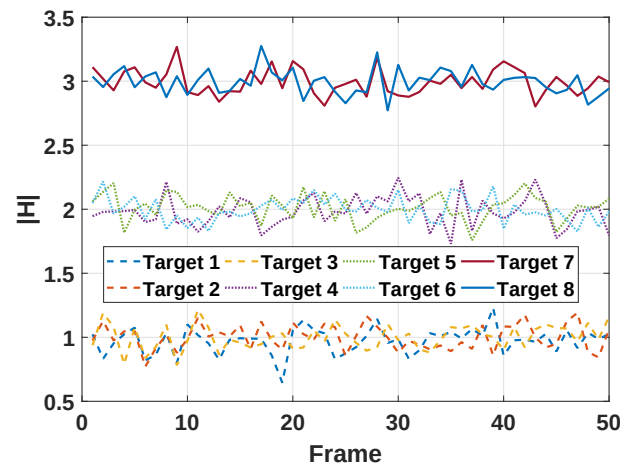


Figure 10. RCS model.

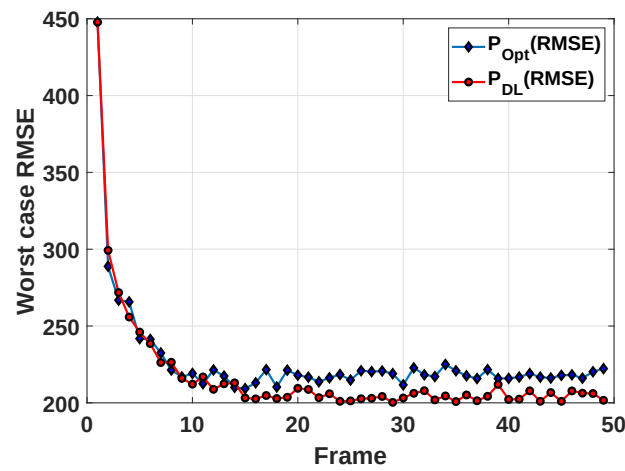


Figure 11. The worst case RMSEs.

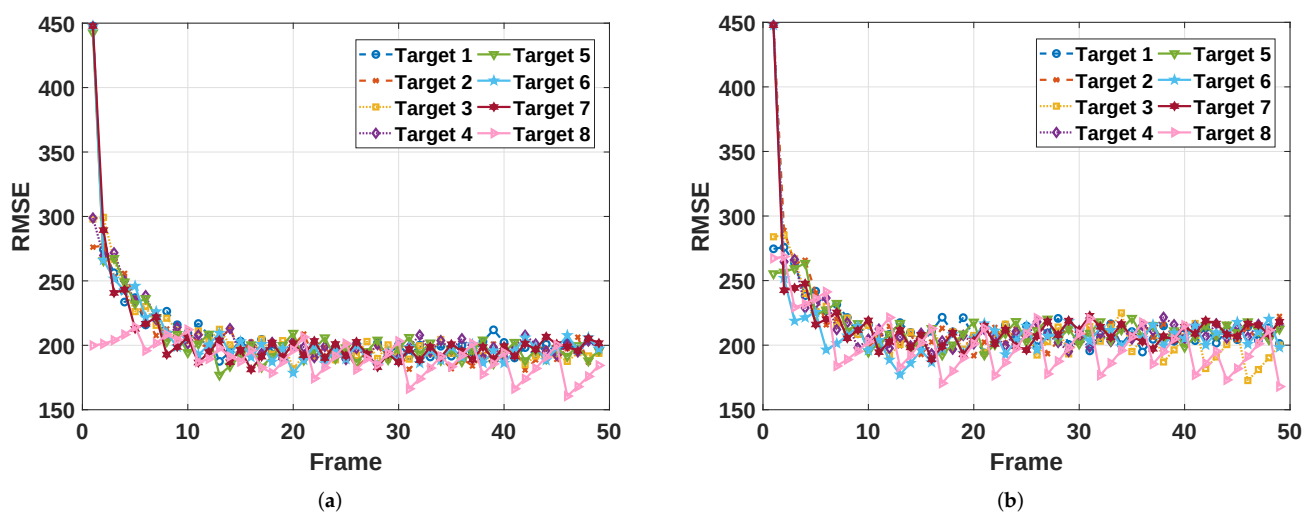


Figure 12. Tracking RMSEs of different targets with given RCS model. (a) Proposed power allocation method. (b) Optimization-based power allocation method.

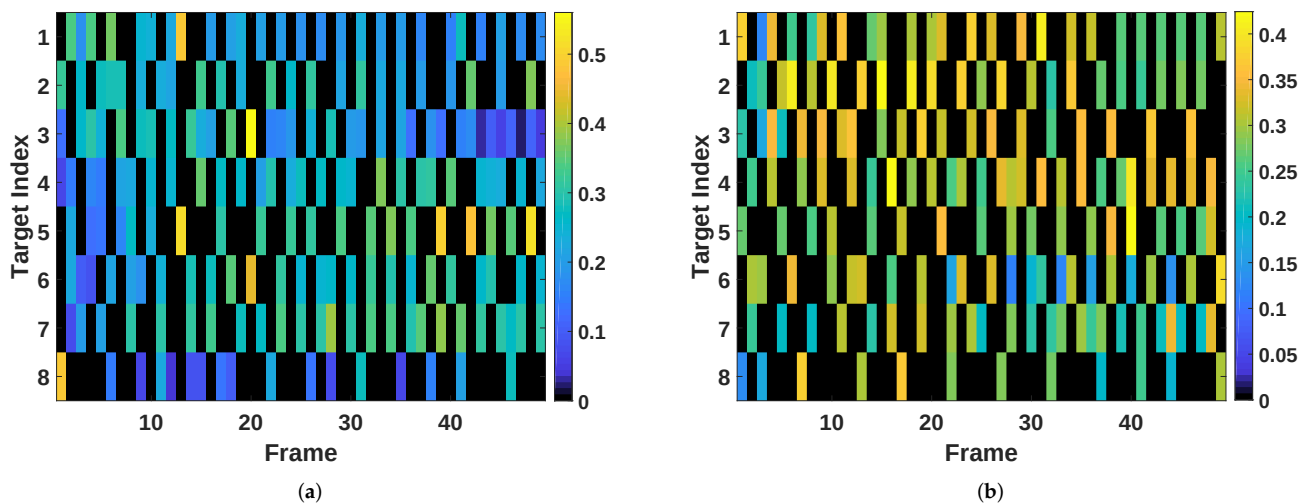


Figure 13. The resource allocation results. (a) Proposed power allocation method. (b) Optimization-based power allocation method.

Simulation 2: In this section, we investigate the effect of the discounted factor γ on the resource allocation results and the tracking performance. Theoretically, taking cumulative discounted tracking accuracy into the resource allocation problem is more reasonable, especially as maneuvering targets are involved. Without loss of generality, three conditions are discussed, i.e., $\gamma = 0$, $\gamma = 0.5$ and $\gamma = 0.99$. The target parameters are set the same as that in simulation 1 and the amplitude of the target reflectivity is also set to 1.

The worst-case tracking RMSEs with different γ are shown in Figure 14, where the result with $\gamma = 0.99$ is lower than that with $\gamma = 0$ and $\gamma = 0.5$ at the beginning, and later surpasses them with frame number increasing. This phenomenon is consistent with that the discounted factor controls the importance of the future tracking accuracy on the current resource allocation solution. Particularly, if taking a small discounted factor, the learned policy tends to improve the performance of the nearer future tracking time step at a cost of a performance loss of the farther future tracking time. In Figure 15, the tracking RMSE curves with $\gamma = 0.5$ are more compact than that with $\gamma = 0.99$ before frame 30 in order to reduce the worst-case tracking RMSE of the nearer future tracking time, which is the opposite after frame 30. The corresponding resource allocation results are presented in Figure 16. In Figure 16a, more power and beam resource are allocated to the beam pointing to target 4, target 5, target 6, and target 7, which are moving away from radar such that a larger gain of future tracking performance can be achieved. Meanwhile, increasing the revisit frequency of Target 3 aims to preventing from future larger BCRLB due to its larger radial speeds, which is the same as the result of $\gamma = 0$. By comparing the tracking performance curves in Figure 14, it is easy to see that the tracking RMSE curves of non-zero discounted factors are lower than that of $\gamma = 0$ within frame [30, 50], which indicates that the usage of non-zero discounted factors makes the resource allocation policy improve the performance of farther tracking time. In this case, as shown in Figure 16b, in order to improve the worst-case tracking RMSE of farther tracking time as $\gamma = 0.99$, Target 7 (it moves away from the radar with the largest radial speed) is illuminated all the time even at the beginning of tracking, and takes up most of the power at the end of tracking. This phenomenon is different from the results with $\gamma = 0$ that cares more about nearer future tracking performance.

In all, the usage of a larger γ contributes to a more stable and effective allocation policy in the case of considering farther future tracking performance. However, in this case, a performance loss of the early tracking stage would not be avoided. In other words, there is a trade-off between the current tracking performance and the future tracking performance about γ .

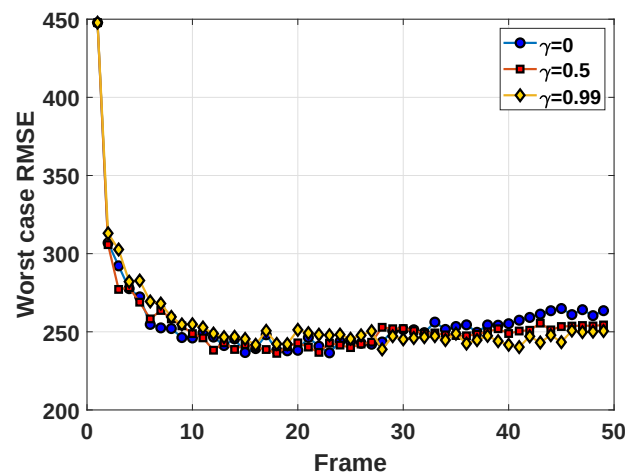


Figure 14. The worst case RMSEs with different γ .

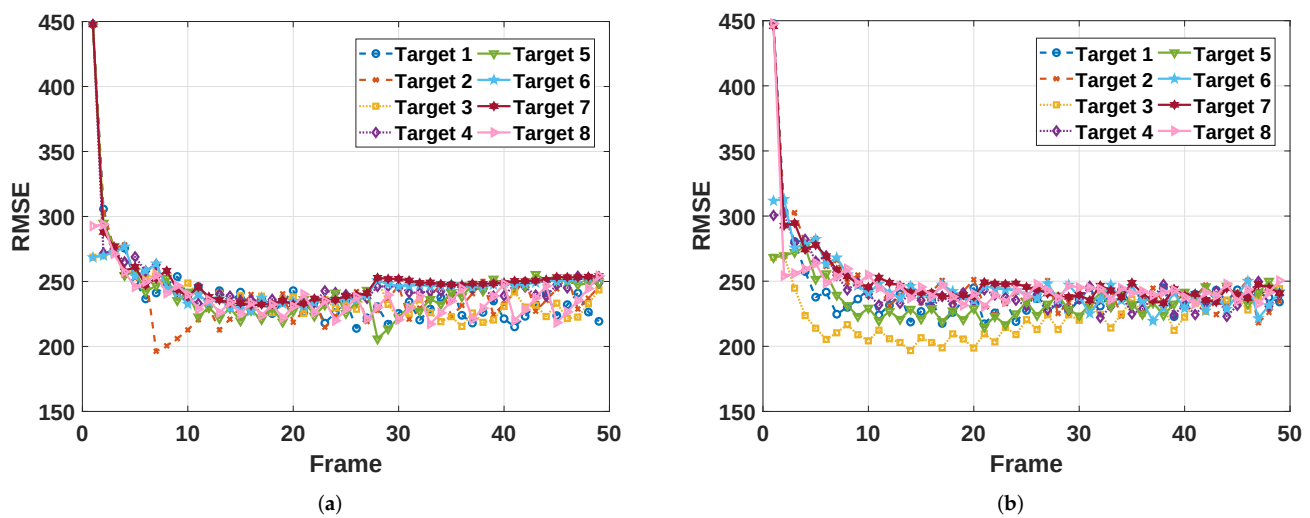


Figure 15. Tracking RMSEs of different targets. (a) $\gamma = 0.5$ (b) $\gamma = 0.99$.

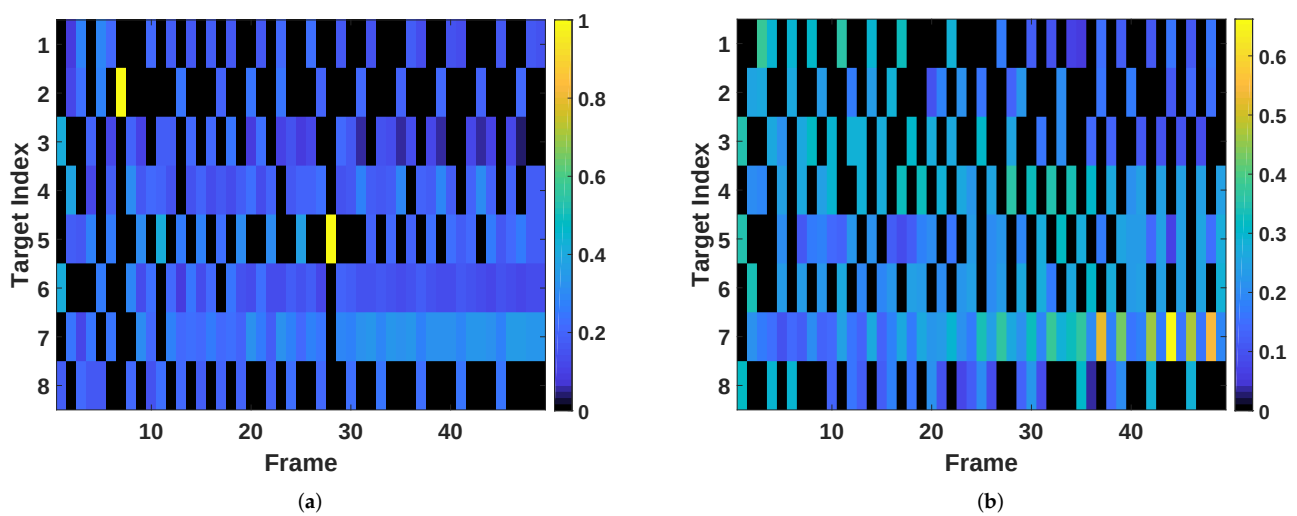


Figure 16. The resource allocation results. (a) $\gamma = 0.5$ (b) $\gamma = 0.99$.

6. Discussion

This paper focuses on the problem of radar power allocation for multi-target tracking from a new perspective. A novel cognitive resource allocation framework is established based on deep learning. Experimental results show that the proposed method is suitable for applications that need to balance the long-term and short-term multi-target tracking performance under limited resources. In practical applications, such as tracking aircraft or missiles, targets often maneuver during tracking process, and the motion models are always unknown due to uncertainty of target motion [8,31]. In this case, the usual way is to use alternative models [32,33], such as CV and CA, for target information prediction including target position, RCS and BCRLB. However, there is a large prediction error in the prediction stage due to the existing model mismatch, which will reduce the resource allocation performance and target tracking performance. Compared with the model-optimization-based methods, the proposed data-driven method can avoid the usage of an alternative model in the prediction stage, and provide more reliable target information for resource allocation. The adoption of the combination of RNN and DRL is able to make the radar system more flexible, intelligent and forward-looking. On the other hand, since the deep-learning-based method does not need to have specific optimization methods for specific problems, we believe that this kind of data-driven method is able to solve most types of radar resource allocation problems in target tracking, such as resource allocation in scenarios of multiple maneuvering target tracking, the resource allocation considering short-term tracking effect or long-term tracking effect, and resource allocation for multiple radar system (MRS).

Our future work will mainly focus on following two aspects:

- (1) Establishing a richer maneuvering dataset. In the simulation experiments of this work, in addition to the uniform motion, the motion dataset of the prediction network also contains the data of some common maneuvering models, such as left-turn motion, right-turn motion, uniform acceleration motion, and so on. However, in real applications, the target maneuvering pattern could be more complex, and there may be the problem of target missing and the association of multiple targets. Therefore, the establishment of a more complete maneuvering target dataset is very important for a cognitive radar system.
- (2) Integrating various types of resource allocation networks and establishing a more intelligent resource allocation system. In theory, the cognitive resource allocation framework based on deep learning proposed in this paper can be used for resource allocation in many scenarios. However, resource allocation networks still need to be trained for different scenarios and different requirements, respectively, such as the scenarios with different target numbers and radar numbers. In practical applications, scenarios and requirements may change rapidly, and thus, the resource allocation strategy needs to be adjusted in real time. How to make the radar system intelligently select the appropriate resource allocation network with the changing of the scenarios and the requirements is of great significance. In our future work, the aim is to unify various types of resource allocation networks to establish an organic system and make the resource allocation process more intelligent.

7. Conclusions

In this paper, a data-driven JBSPA method for MTT is proposed and the simulation results have demonstrated its superiority. There are two keys to enhancing the MTT performance and the resource utilization efficiency, one is using the LSTM-GMM for target information prediction, which provides a reliable prior information for resource allocation, the other one is forming an MDP of resource allocation and solving with the DRL, which realizes complicate resource allocation considering long-term tracking performance. In summary, this method integrates the decision-making ability of DRL and the memory and prediction ability of RNN to make the resource allocation more flexible and further improve the cognitive ability of the radar system.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S.; software, Y.S. and H.Z.; validation, Y.S. and K.L.; investigation, H.Z.; writing—original draft preparation, Y.S.; writing—review and editing, K.L.; supervision, Y.S. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B18039), Shaanxi Innovation Team Project.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Blackman, S. *Multiple-Target Tracking with Radar Applications*; Artech House: Norwood, MA, USA, 1986.
- Stone, L.D.; Barlow, C.A.; Corwin, T. *Bayesian Multiple Target Tracking*; Artech House: Norwood, MA, USA, 1999.
- Hue, C.; Cadre, J.L.; Pérez, P. Sequential Monte Carlo methods for multiple target tracking and data fusion. *IEEE Trans. Signal Process.* **2002**, *50*, 309–325. [[CrossRef](#)]
- Stoica, P.; Li, J. MIMO radar with colocated antennas. *IEEE Signal Process. Mag.* **2007**, *24*, 106–114.
- Gorji, A.A.; Tharmarasa, R.; Blair, W.D.; Kirubarajan, T. Multiple unresolved target localization and tracking using colocated MIMO radars. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2498–2517. [[CrossRef](#)]
- Rabideau, D.J.; Parker, P. *Ubiquitous MIMO Multifunction Digital Array Radar and the Role of Time-Energy Management in Radar*; Project Rep. DAR-4; MIT Lincoln Lab.: Lexington, MA, USA, 2003.
- Godrich, H.; Petropulu, A.P.; Poor, H.V. Power allocation strategies for target localization in distributed multiple-radar architecture. *IEEE Trans. Signal Process.* **2011**, *59*, 3226–3240. [[CrossRef](#)]
- Chavali, P.; Nehorai, A. Scheduling and power allocation in a cognitive radar network for multiple-target tracking. *IEEE Trans. Signal Process.* **2012**, *50*, 715–729. [[CrossRef](#)]
- Yan, J.; Dai, J.; Pu, W.; Zhou, S.; Liu, H.; Bao, Z. Quality of service constrained-resource allocation scheme for multiple target tracking in radar sensor network. *IEEE Syst. J.* **2021**, *15*, 771–779. [[CrossRef](#)]
- Haykin, S.; Zia, A. Arasaratnam, I.; Yanbo, X. Cognitive tracking radar. In Proceedings of the 2010 IEEE Radar Conference, Washington, DC, USA, 10–14 May 2010.
- Tichavsky, P.; Muravchik, C.H.; Nehorai, A. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Trans. Signal Process.* **1998**, *46*, 1386–1396. [[CrossRef](#)]
- Hernandez, M.; Kirubarajan, T.; Bar-Shalom, Y. Multisensor resource deployment using posterior Cramér-Rao bounds. *IEEE Trans. Aerosp. Electron. Syst.* **2004**, *40*, 399–416. [[CrossRef](#)]
- Hernandez, M.L.; Farina, A.; Ristic, B. PCRLB for tracking in cluttered environments: Measurement sequence conditioning approach. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 680–704. [[CrossRef](#)]
- Yan, J.; Liu, H.; Jiu, B.; Chen, B.; Liu, Z.; Bao, Z. Simultaneous multibeam resource allocation scheme for multiple target tracking. *IEEE Trans. Signal Process.* **2015**, *63*, 3110–3122. [[CrossRef](#)]
- Shi, Y.; Jiu, B.; Yan, J.; Liu, H.; Wang, C. Data-driven simultaneous multibeam power allocation: When multiple targets tracking meets deep reinforcement learning. *IEEE Syst. J.* **2021**, *15*, 1264–1274. [[CrossRef](#)]
- Cormode, G.; Thottan, M. *Algorithms for Next Generation Networks*; Springer: London, UK, 2010.
- Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
- Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [[CrossRef](#)] [[PubMed](#)]
- Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
- Shi, Y.; Jiu, B.; Yan, J.; Liu, H. Data-Driven radar selection and power allocation method for target tracking in multiple radar system. *IEEE Sens. J.* **2021**, *21*, 19296–19306. [[CrossRef](#)]
- Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
- Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic policy gradient algorithms. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014.
- Sutton, R.S.; McAllester, D.A.; Singhand, S.P.; Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In Proceedings of the 12th International Conference on Neural Information Processing Systems, Denver, CO, USA, 27–30 November 2000.
- Van Trees, H.L. *Detection, Estimation, and Modulation Theory, Part III*; Wiley: New York, NY, USA, 1971.
- Van Trees, H.L. *Optimum Array Processing: Detection, Estimation, and Modulation Theory, Part IV*; Wiley: New York, NY, USA, 2002.
- Niu, R.; Willett, P.; Bar-Shalom, Y. Matrix CRLB scaling due to measurements of uncertain origin. *IEEE Trans. Signal Process.* **2001**, *49*, 1325–1335.

27. Zhang, X.; Willett, P.; Bar-Shalom, Y. Dynamic Cramer-Rao bound for target tracking in clutter. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 1154–1167. [[CrossRef](#)]
28. Stocia, P.; Selén, Y. Cyclic minimizer, majorization techniques, and expectation-maximization algorithm: A refresher. *IEEE Signal Process. Mag.* **2004**, *21*, 112–114.
29. Kingma D.; Ba, J. Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference for Design, Santiago de Chile, Chile, 28–30 June 2015.
30. Ristic, B.; Arulampalam, S.; Gordon, N. *Beyond the Kalman Filter Particle Filters for Tracking Applications*; Artech House: Norwell, MA, USA, 2004.
31. Yan, J.; Pu, W.; Liu, H.; Bo, J.; Bao, Z. Robust chance constrained power allocation scheme for multiple target localization in colocated MIMO system. *IEEE Trans. Signal Process.* **2018**, *21*, 3946–3957. [[CrossRef](#)]
32. Yan, J.; Pu, W.; Zhou, S.; Liu, H.; Bao, Z. Collaborative detection and power allocation framework for target tracking in multiple radar system. *Inf. Fusion* **2020**, *55*, 178–183. [[CrossRef](#)]
33. Yan, J.; Pu, W.; Zhou, S.; Liu, H.; Greco, M.S. Optimal Resource Allocation for Asynchronous Multiple Targets Tracking in Heterogeneous Radar Network. *IEEE Trans. Signal Process.* **2020**, *68*, 4055–4068. [[CrossRef](#)]