TYPE Original Research
PUBLISHED 24 February 2023
DOI 10.3389/fnbot.2023.1093132

Check for updates

OPEN ACCESS

EDITED BY
Yimin Zhou,
Shenzhen Institutes of Advanced Technology (CAS), China

REVIEWED BY
Huaming Qian,
Harbin Engineering University, China
Tim Verbelen,
Ghent University, Belgium
Vittorio Cuculo,
University of Milan, Italy

*CORRESPONDENCE
Feng Zhu
✉ fzhu@sia.cn

RECEIVED 08 November 2022
ACCEPTED 08 February 2023
PUBLISHED 24 February 2023

CITATION
Sun H, Zhu F, Li Y, Zhao P, Kong Y, Wang J, Wan Y and Fu S (2023) Viewpoint planning with transition management for active object recognition. *Front. Neurorobot.* 17:1093132. doi: 10.3389/fnbot.2023.1093132

Viewpoint planning with transition management for active object recognition

Haibo Sun[1,2,3,4], Feng Zhu[2,3,4]*, Yangyang Li[2,3,4,5], Pengfei Zhao[2,3,4,5], Yanzi Kong[2,3,4,5], Jianyu Wang[1,2,3,4], Yingcai Wan[1] and Shuangfei Fu[2,3,4]

[1]Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China, [2]Key Laboratory of Opto-Electronic Information Processing, Chinese Academy of Sciences, Shenyang, China, [3]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, [4]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang, China, [5]University of Chinese Academy of Sciences, Beijing, China

Active object recognition (AOR) provides a paradigm where an agent can capture additional evidence by purposefully changing its viewpoint to improve the quality of recognition. One of the most concerned problems in AOR is viewpoint planning (VP) which refers to developing a policy to determine the next viewpoints of the agent. A research trend is to solve the VP problem with reinforcement learning, namely to use the viewpoint transitions explored by the agent to train the VP policy. However, most research discards the trained transitions, which may lead to an inefficient use of the explored transitions. To solve this challenge, we present a novel VP method with transition management based on reinforcement learning, which can reuse the explored viewpoint transitions. To be specific, a learning framework of the VP policy is first established *via* the deterministic policy gradient theory, which provides an opportunity to reuse the explored transitions. Then, we design a scheme of viewpoint transition management that can store the explored transitions and decide which transitions are used for the policy learning. Finally, within the framework, we develop an algorithm based on twin delayed deep deterministic policy gradient and the designed scheme to train the VP policy. Experiments on the public and challenging dataset GERMS show the effectiveness of our method in comparison with several competing approaches.

KEYWORDS

active object recognition, viewpoint planning, deterministic policy gradient, twin delayed deep deterministic policy gradient, viewpoint transition management, reinforcement learning

1. Introduction

Visual object recognition has a wide range of applications e.g., automatic driving (Behl et al., 2017), robotics (Stria and Hlavác, 2018), medical diagnostic (Duan et al., 2019), environmental perception (Roynard et al., 2018), etc. Most recognition systems merely take a single viewpoint image as input and produce a category label estimate as output (Jayaraman and Grauman, 2019). It is prone to the recognition errors when the image can not provide sufficient information. In contrast, the visual behavior of people is an active process so as to more clearly perceive their surroundings. As shown in Figure 1, in daily life, people can intelligently observe an object from different viewpoints to determine the identity of the object. Similarly, if the viewpoint of an agent can be adjusted (e.g., mobile robots and

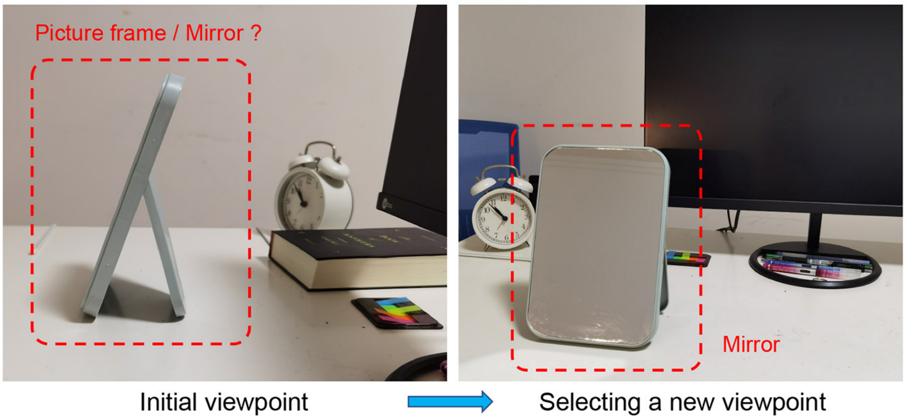

FIGURE 1
An example illustrating the active preception process of people.

autonomous vehicles), more valuable information will be obtained to boost the recognition performance.

As a branch of active vision (Parr et al., 2021), active object recognition (AOR) (Patten et al., 2015; Wu et al., 2015; Potthast et al., 2016; Van de Maele et al., 2022) is a typical technology to realize the above idea, which aims to collect additional clues by purposefully changing the viewpoint of an agent to improve the quality of recognition. Andreopoulos and Tsotsos (2013) and Zeng et al. (2020) review a series of classical AOR methods. One of the most concerned problems in AOR is viewpoint planning (VP) that refers to developing a policy to determine the next viewpoints of the agent. In recent years, researchers mainly focus on using reinforcement learning to solve the VP problem (Becerra et al., 2014; Malmir et al., 2015; Malmir and Cottrell, 2017; Liu et al., 2018a), namely to use the viewpoint transitions explored by the agent to train the VP policy. Becerra et al. (2014) formally define object recognition as a partially observable Markov decision process problem and uses stochastic dynamic programming to address the problem. As a pioneering work, Malmir et al. (2015) provide a public AOR dataset called GERMS that includes 136 objects with different view images and develops a deep Q-learning (DQL) system to learn to actively verify objects by using standard back-propagation and Q-learning. In the same way, Liu et al. (2018a) design a hierarchical local-receptive-field architecture to predict object label and learns a VP policy by combining extreme learning machine and Q-learning. Similar to Becerra et al. (2014), AOR is also modeled as a partially observable Markov decision process by Malmir and Cottrell (2017). The difference is that a belief tree search is built to find near-optimal action values which correspond to the next best viewpoints. These VP methods explore discrete viewpoint space, which may introduce significant quantization errors. Hence, Liu et al. (2018b) present a continuous VP method based on trust region policy optimization (TRPO) (Schulman et al., 2015) and adopts extreme learning machine (Huang et al., 2006) to reduce computational complexity. It shows a promising result on the GERMS dataset compared to the discrete VP methods. However, due to the on-policy characteristic of TRPO, the trained viewpoint transitions will be discarded by the agent, which may lead to an inefficient use of the explored transitions.

The deterministic policy gradient theory (Silver et al., 2014) is proposed for reinforcement learning with continuous actions and introduces an off-policy actor-critic algorithm (OPDAC-Q) to learn a deterministic target policy. Lillicrap et al. (2015) present a deep deterministic policy gradient (DDPG) approach that combines deterministic policy gradient with DQN (Mnih et al., 2013, 2015) to learn policies in high-dimensional continuous action spaces. Fujimoto et al. (2018) contribute a mechanism that takes the minimum value between a pair of critics in the actor-critic algorithm of Silver et al. (2014) to tackle the function approximation errors. The deterministic policy gradient theory has been widely applied in various fields, such as electricity market (Liang et al., 2020), vehicle speed tracking control (Hao et al., 2021), fuzzy PID controller (Shi et al., 2020), quadrotor control (Wang et al., 2020), energy efficiency (Zhang et al., 2020), and autonomous underwater vehicles (Sun et al., 2020; Wu et al., 2022). However, to our best knowledge, it has never been employed in the AOR task.

In this work, we present a novel continuous VP method with transition management based on reinforcement learning. This method can efficiently use the explored viewpoint transitions to learn the continuous VP policy. Concretely, a learning framework of the continuous VP policy is established using the deterministic policy gradient theory, which provides an opportunity to reuse the explored transitions owing to the off-policy characteristic of the theory. Then, we design a scheme of viewpoint transition management that can store the explored transitions and decide which transitions are used for the policy learning. The scheme is implemented by introducing and improving the prioritized experience replay technology (Schaul et al., 2016). The improvements include: (1) We improve the estimation approach of temporal difference (TD) error with the clipped double Q-learning algorithm (Fujimoto et al., 2018) so as to adapt to our continuous VP framework. (2) We utilize importance-sampling to correct the estimation bias of TD error produced by the prioritized replay. Finally, within the framework, we develop an algorithm based on twin delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018) and the designed scheme to train the continuous VP policy. Experimental results on the public dataset GERMS demonstrate the effectiveness of the proposed VP method.

The key contributions of this work are

- A novel continuous VP method with transition management for AOR is presented to solve the problem of inefficient use of the explored viewpoint transitions in the existing continuous VP method.
- We establish a learning framework of the continuous VP policy *via* the deterministic policy gradient theory.
- A scheme of viewpoint transition management is designed, which is implemented by introducing and improving the prioritized experience replay technology.
- We develop an algorithm based on twin delayed deep deterministic policy gradient and the designed scheme to train the continuous VP policy.

The rest of this paper is structured as follows: Section 2 formulates the VP problem. Section 3 details the proposed framework for the solution of the problem. Finally, the implementation and experimental results, as well as conclusions are further provided in Sections 4, 5.

2. Problem definition

An AOR system mounted on an automatic mobile agent allows the agent to identify an object by dealing with the images captured from different viewpoints. Suppose at the initial time $t = 0$, an object to be identified is given from an object library containing M objects and the agent captures an image I_{Φ_0} from the initial viewpoint Φ_0. The classifier $\mathcal{C}(\cdot)$ in the AOR system will give a probability prediction $\mathcal{C}(I_{\Phi_0})$ of the object according to the image I_{Φ_0}. $\mathcal{C}(I_{\Phi_0})$ is a M dimensional vector where every element denotes recognition probability of different objects in the library. When the prediction is uncertain [i.e., the maximum probability in $\mathcal{C}(I_{\Phi_0})$ is less than the preset threshold], the agent will move to explore more viewpoints to improve recognition performance. This requires the system plans a relative movement action a_t for the agent to obtain a new viewpoint $\Phi_{t+1} = \Phi_t + a_t$. The new image $I_{\Phi_{t+1}}$ captured from the viewpoint Φ_{t+1} will be used for the recognition again. This process is repeated several times until a stop condition (e.g., planning up to T_{max} time steps or reaching the preset probability threshold) is reached.

An undesirable planning action may make it difficult for the agent to capture useful images for recognition. Therefore, we need to find an effective VP policy for the AOR system. For this purpose, the VP problem is considered as a reinforcement learning paradigm which can be formulated as a Markov decision process. The process is described with a six-element tuple $< S, A, r, \mathcal{P}, \gamma, u >$.

- S represents a set of continuous states in which each state s is produced by the predictions of corresponding images captured from different viewpoints.
- A is a set of continuous actions which are determined by the agent. Each action a in the set is used for the agent to get a new viewpoint.
- $r: S\times A\rightarrow\mathbb{R}$ is a reward function designed to evaluate the quality of selecting a viewpoint.
- $\mathcal{P}: S\times A\times S\rightarrow[0,1]$ denotes the transition probability. It describes the possibility of transferring to the subsequent state s' after the action a is selected in the state s.
- $\gamma \in [0,1]$ is a discount factor used to adjust the attention between present and future rewards.
- $u: S\rightarrow A$ is a deterministic continuous VP policy [i.e., $a = u(s)$] that can generate an action for the agent to get a new viewpoint in a certain state.

The VP problem is transformed to solve the optimal policy u^* in the setting of reinforcement learning.

3. Method

3.1. Overview

In reinforcement learning, the optimal policy u^* can be achieved by maximizing the expected return over all episodes. At any time step t of each episode, with a given state $s_t\in S$, the agent plans an action $a_t\in A$ according to its current policy u ($a_t = u(s_t)$), receiving a reward $r(s_t, a_t)$ and the new state $s_{t+1}\sim\mathcal{P}(s_{t+1}|s_t, a_t)$. ($(s_t, a_t, r_t, s_{t+1})$ is called the viewpoint transition in the AOR task.) The return is defined as the cumulative discounted reward $\sum_{i=t}^{T}\gamma^{i-t}r(s_i, a_i)$ where T is the end time step of planning. Let $Q^u(s_t, a_t)$ be the expected return when performing action a_t in state s_t under the policy u. $Q^u(s_t, a_t)$ is defined as

$$Q^u(s_t, a_t) = \mathop{\mathbb{E}}_{s_{t+1}\sim\mathcal{P}(s_{t+1}|s_t,a_t)}[\sum_{i=t}^{T}\gamma^{i-t}r(s_i, a_i)|s_t, a_t] \quad (1)$$

which is known as the action value function. u^* can be solved by maximizing the expected value of Equation (1) over the whole state space

$$u^* = \max_u \mathbb{E}_{s_t\sim d(\cdot)}[Q^u(s_t, a_t)|a_t = u(s_t)] \quad (2)$$

where $d(\cdot)$ is the state probability density of Markov decision process in steady state distribution (Bellemare et al., 2017).

We assume the deterministic continuous VP policy u is parameterized by θ and denote it as $u(s;\theta)$. Naturally, Equation (2) can be transformed to an optimization with respect to θ that maximize the objective

$$J(\theta) = \mathbb{E}_{s_t\sim d(\cdot)}[Q^u(s_t, a_t)|a_t = u(s_t;\theta)]. \quad (3)$$

To solve the optimization of Equation (3), the deterministic policy gradient theory (Silver et al., 2014) is introduced to iteratively update the parameters θ by taking the gradient of Equation (3)

$$\nabla_\theta J(\theta) = \mathbb{E}_{s_t\sim d(\cdot)}[\nabla_\theta u(s_t;\theta)\nabla_a Q^u(s_t, a_t)|a_t = u(s_t;\theta)]. \quad (4)$$

We utilize (Equation 4) as a framework to learn the optimal deterministic continuous VP policy $u(s_t;\theta^*)$ for AOR. The reason why this framework can reuse the explored viewpoint transitions is the off-policy characteristic of the deterministic policy gradient

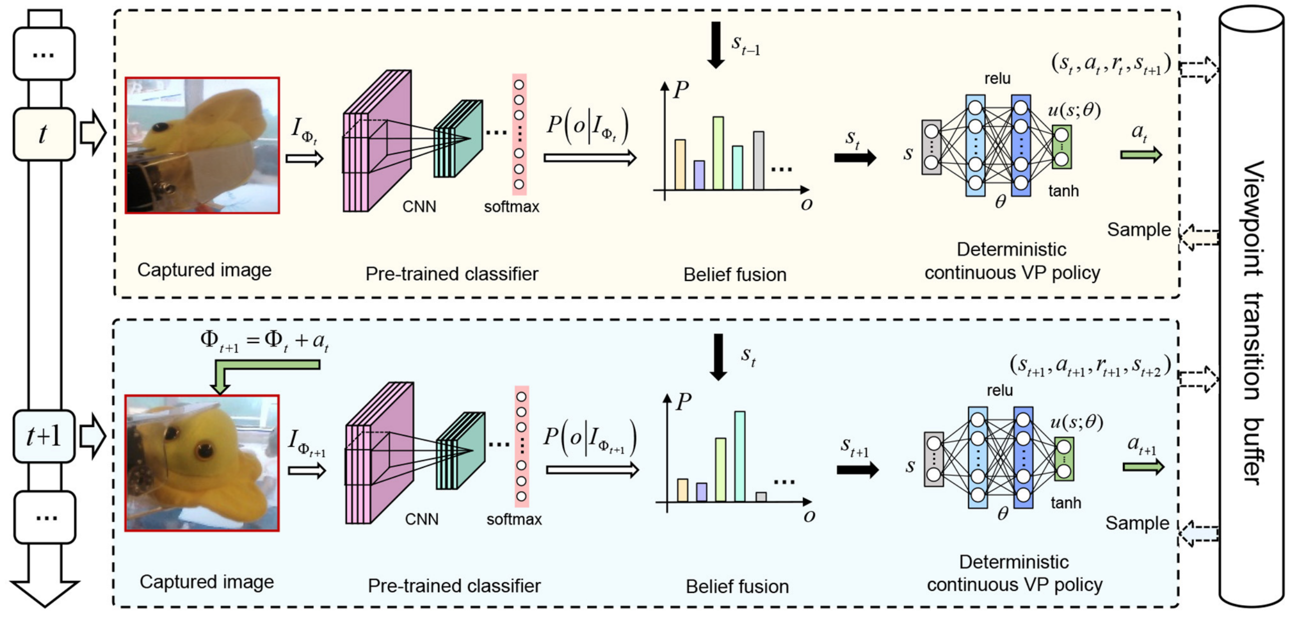

FIGURE 2
The pipeline of active object recognition based on deterministic continuous viewpoint planning. The deterministic policy gradient theory (Silver et al., 2014) is introduced to build a framework of continuous viewpoint planning. We design a scheme of viewpoint transition management to store and replay the explored viewpoint transitions. Within the framework, we develop an algorithm based on TD3 (Fujimoto et al., 2018) and the scheme to train the VP policy network. During the training, the agent stores the explored viewpoint transition (s_t, a_t, r_t, s_{t+1}) in the viewpoint transition buffer and samples a mini-batch transitions from it to train the VP policy network at each time step.

theory, i.e., the viewpoint transitions explored by any policy can be used for the calculation of the gradient in Equation (4), because the gradient is only related to the distribution of state s_t (Silver et al., 2014). The pipeline of our AOR is shown in Figure 2 where the VP policy $u(s_t;\theta)$ is represented by a three-layer fully-connected neural network with the parameters θ. The policy network $u(s_t;\theta)$ takes a state s_t as input and outputs a deterministic action $a_t = u(s_t;\theta)$. In the following, the representations of state s_t and reward function $r(s_t, a_t)$ will be elaborated. Additionally, we will design a scheme of viewpoint transition management and develop a training algorithm based on twin delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018) and the scheme for the learning of $u(s_t;\theta^*)$ within the framework.

3.2. Recognition state

As shown in Figure 2, we first use a convolutional neural network (CNN) model to extract features from the captured image I_{Φ_t} and then recognize the concerned objects with a *softmax* layer added the top of the CNN model. The CNN model and the *softmax* layer constitute a classifier $\mathcal{C}(\cdot)$ which is pre-trained with the images from different viewpoints of the concerned objects. The parameters of the classifier are fixed when training the VP policy network. The classifier outputs a belief vector $\mathcal{C}(I_{\Phi_t})$ where every element denotes recognition probability of different objects. The *oth* element in the vector is represented as $P(o|I_{\Phi_t})$ where $o = 1, 2, ..., M$ is the object label. The recognition state s_t is a posterior probability distribution over different objects at time step t, which is produced by the captured images. It is also expressed as a vector where the *oth* element is $P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_t}), o = 1, 2, ..., M$. According to naive Bayes (Paletta and Pinz, 2000), $P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_t})$ is given as

$$\xi_t P(o|I_{\Phi_t})P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_{t-1}}) \tag{5}$$

where ξ_t is a normalizing coefficient.

3.3. Reward function

Reward function $r(s_t, a_t)$ (denoted as r_t for simplicity) is used to evaluate the quality of selecting a viewpoint. As described in Section 3.2, state is a posterior probability distribution over different objects. The flatter the distribution is, the stronger the recognition uncertainty is. To quantify the uncertainty, information entropy (Zhao et al., 2016; Liu et al., 2018b) is utilized and the uncertainty in state s_t is denoted as $H(s_t) = -\sum_o P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_t}) \log P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_t})$. The purpose of AOR is to reduce the uncertainty of recognition through viewpoint planning. Therefore, we can design the reward function according to the change of uncertainty before and after viewpoint selection. The resulting reward function is

$$r_t = \begin{cases} -1, & \hat{o}_{t+1} \neq o^* \\ 0, & \hat{o}_{t+1} = o^*, H(s_{t+1}) \geq H(s_t) \\ 1, & \hat{o}_{t+1} = o^*, H(s_{t+1}) < H(s_t) \end{cases} \tag{6}$$

where o^* is the object label and $\hat{o}_{t+1} = argmax_o P(o|I_{\Phi_0}, I_{\Phi_1}, ..., I_{\Phi_{t+1}})$ is the predicted result. When the predicted result is right ($\hat{o}_{t+1} = o^*$) and the uncertainty is reduced ($H(s_{t+1}) < H(s_t)$), it indicates that this viewpoint selection

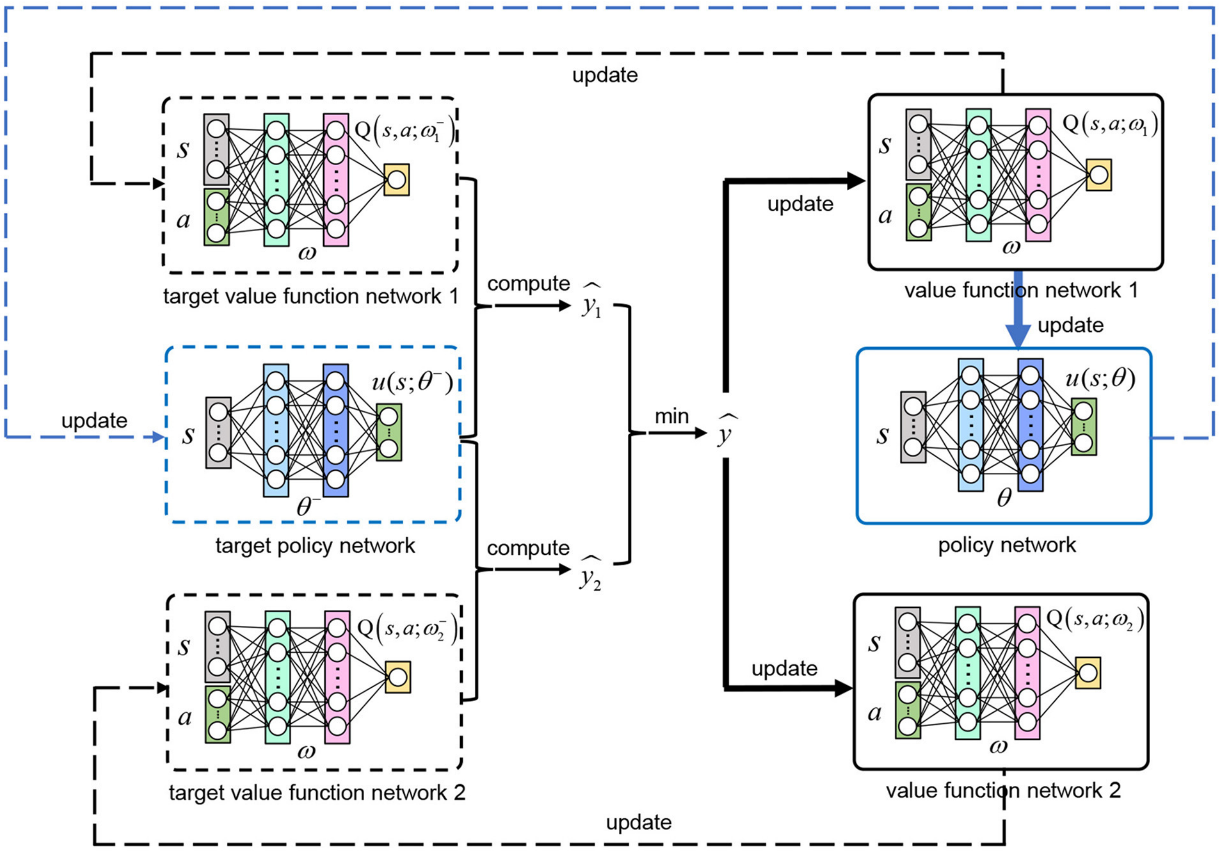

FIGURE 3
The relationship between the six networks. The TD target $\hat{y}$ is estimated with the target value function network 1 and 2 using our clipped double Q-learning and bias correction based algorithm (Equation 12), which is used to update the value function network 1 and 2. With the gradient of $Q(s_t, a_t; \omega_1)$ to a, the policy network is updated with Equation (13). Three target networks ($u(s_t; \theta^-)$, $Q(s_t, a_t; \omega_1^-)$, $Q(s_t, a_t; \omega_2^-)$) adopt soft updates according to their corresponding evaluation networks ($u(s_t; \theta)$, $Q(s_t, a_t; \omega_1)$, $Q(s_t, a_t; \omega_2)$).

is valuable for recognition. On the contrary, other situations mean that this viewpoint selection is not good.

3.4. Viewpoint transition management

The agent can obtain a transition (s_t, a_t, r_t, s_{t+1}) after a viewpoint selection and use it for the learning of the continuous VP policy. In the TRPO-based VP method (Liu et al., 2018b), the obtained viewpoint transitions will be discarded after they are trained due to the on-policy characteristic of TRPO. It leads to a low efficient use of the obtained transitions. In our work, the deterministic policy gradient theory (Silver et al., 2014) allows the agent to reuse the obtained transitions. Therefore, to make full use of the obtained viewpoint transitions, the experience replay (ER) (Lin, 1992; Schaul et al., 2016) technology is adopted and improved to implement a scheme of viewpoint transition management. The scheme includes viewpoint transition storage and viewpoint transition reuse.

3.4.1. Viewpoint transition storage

To store the obtained viewpoint transitions, we build a viewpoint transition buffer with a capacity of K in the light of Lin (1992) and Schaul et al. (2016). K is generally within $10^4 \sim 10^6$. Once the buffer is full of transitions, the old ones will be replaced by the newly generated transitions.

3.4.2. Viewpoint transition reuse

The key of viewpoint transition reuse is to decide which transitions to reuse. Lin (1992) adopt a uniform sampling strategy that means the sampling probability of each transition in the buffer is the same. However, those transitions with greater temporal difference (TD) errors are obviously more surprising to the agent and should be sampled with a higher probability (Schaul et al., 2016). Hence, Schaul et al. (2016) present a prioritized experience replay (PER) technology that can quantify the surprising level (priority) of each transition by the TD error and convert the priority into the corresponding sampling probability. Here, we employ the PER technology to sample the viewpoint transitions in the buffer. Concretely, the probability of sampling the ith stored viewpoint transition is given as

$$P(i) = \frac{p_i^\lambda}{\sum_{l=1}^{K} p_l^\lambda} \tag{7}$$

where $p_i^\lambda > 0$ is the priority of the ith transition. The exponent λ indicates how much prioritization is used, with $\lambda = 0$

corresponding to the uniform case. Proportional prioritization is defined with

$$p_i = |\hat{\delta}_i| + \epsilon \tag{8}$$

where $\hat{\delta}_i$ is the TD error of the ith transition and ϵ is a small positive value that prevents transitions with error of 0 from not being sampled. The estimation of TD error in PER is based on the double DQN algorithm (Mnih et al., 2015).

$$\hat{\delta}_i = r_t^{(i)} + \gamma Q(s_{t+1}^{(i)}, argmax_a Q(s_{t+1}^{(i)}, a; \omega); \omega^-) - Q(s_t^{(i)}, a_t^{(i)}; \omega) \tag{9}$$

where $Q(s_t, a_t; \omega)$ and $Q(s_t, a_t; \omega^-)$ are value function network and target value function network respectively. However, it is only applicable to discrete viewpoint planning, not to our continuous case. Inspired by Fujimoto et al. (2018), we improve the estimation method of TD error with the clipped double Q-learning algorithm so as to adapt to our deterministic continuous VP framework. The improved TD error is

$$\hat{\delta}_i = |\hat{y}_t^{(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_1)| + |\hat{y}_t^{(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_2)| \tag{10}$$

where $\hat{y}_t^{(i)} = r_t^{(i)} + \gamma \min_{j=1,2} Q(s_{t+1}^{(i)}, u(s_{t+1}^{(i)}; \theta^-); \omega_j^-)$ is TD target. $Q(s_t, a_t; \omega_1)$ and $Q(s_t, a_t; \omega_2)$ are two value function networks, and $Q(s_t, a_t; \omega_1^-)$ and $Q(s_t, a_t; \omega_2^-)$ are their corresponding target value function networks. $u(s_t; \theta^-)$ is the target policy network. These networks will be elaborated in the next subsection.

In addition, we find that the estimation of TD error is biased due to the prioritized sampling. It is known that Bellman optimality equation (Sutton and Barto, 2018) is $Q(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)}[r_t + \gamma \max_a Q(s_{t+1}, a)]$ where $y_t = r_t + \gamma \max_a Q(s_{t+1}, a)$ is TD target. Obviously, the distribution $s_{t+1} \sim \mathcal{P}(s_{t+1}|s_t, a_t)$ is changed by using the prioritized sampling, which introduces bias to the estimation of the expected value $Q(s_t, a_t)$. Thus, we correct the bias with importance-sampling weight $\rho = \frac{\mathcal{P}}{\mathcal{D}}$ where $\mathcal{D}$ is the new distribution of s_{t+1} generated due to the use of prioritized sampling. Then Bellman optimality equation is transformed to $Q(s_t, a_t) = \mathbb{E}_{s_{t+1} \sim \mathcal{D}(s_{t+1}|s_t, a_t)}[\rho(r_t + \gamma \max_a Q(s_{t+1}, a)]$ where $\rho(r_t + \gamma \max_a Q(s_{t+1}, a)$ is TD target with bias correction denoted as y_t^{corr}. And TD error is transformed to $\delta = y_t^{corr} - Q(s_t, a_t)$. Similar, in our scheme, the importance-sampling weight of the *ith* viewpoint transition in the buffer is

$$\rho_i = \frac{1}{K \cdot P(i)} \tag{11}$$

where K is the capacity of the buffer. Our clipped double Q-learning based TD error and TD target are corrected as

$$\begin{aligned} \hat{\delta}_i^{corr} &= |\hat{y}_t^{corr(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_1)| + |\hat{y}_t^{corr(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_2)| \\ \hat{y}_t^{corr(i)} &= \rho_i(r_t^{(i)} + \gamma \min_{j=1,2} Q(s_{t+1}^{(i)}, u(s_{t+1}^{(i)}; \theta^-); \omega_j^-)). \end{aligned} \tag{12}$$

To avoid expensive sweeps over the entire viewpoint transition buffer, priorities are only updated for the transitions that are sampled according to Schaul et al. (2016). In addition, the new transitions will be put in the buffer with maximal priority in order to guarantee that all transitions are seen at least once.

```
Input: Parameters: $\sigma_1, N, \sigma_2, c, \beta, d, \alpha, \tau, K$
Output: $\theta$
Initialize the value function networks
  $Q(s_t, a_t; \omega_1), Q(s_t, a_t; \omega_2)$, and the VP policy network
  $u(s_t; \theta)$ with random parameters $\omega_1, \omega_2, \theta$
Initialize the target networks
  $\omega_1^- \leftarrow \omega_1, \omega_2^- \leftarrow \omega_2, \theta^- \leftarrow \theta$
Initialize the viewpoint transition buffer $\mathcal{B}$
  with the capacity $K$
for $t = 1$ to $T$ do
    Run a behavioral policy with exploration noise
      to select an action $\tilde{a}_t \sim u(s_t; \theta) + \epsilon_1, \epsilon_1 \sim \mathcal{N}(0, \sigma_1)$
      and receive a reward $r_t$ and a new state $s_{t+1}$
    Store the transition tuple $(s_t, \tilde{a}_t, r_t, s_{t+1})$ in $\mathcal{B}$
      with maximal priority
    for $i = 1$ to $N$ do
        Sample transitions $(s_t^{(i)}, \tilde{a}_t^{(i)}, r_t^{(i)}, s_{t+1}^{(i)})$ from the
          buffer $\mathcal{B}$: $i \sim P(i) = \frac{p_i^\lambda}{\sum_{l=1}^K p_l^\lambda}$ (Equation 7)
        Compute importance-sampling weight $\rho_i$
          (Equation 11)
        Estimate the corrected TD targets $\hat{y}_t^{corr(i)}$
          using Equation (12)
        Compute $\tilde{a}_{t+1} = u(s_{t+1}; \theta^-) + \epsilon_2, \epsilon_2 \sim clip(\mathcal{N}(0, \sigma_2), -c, c)$
          according to the smoothing regularization
          of TD3 (Fujimoto et al., 2018)
        Estimate the corrected TD error $\hat{\delta}_i^{corr}$
          (Equation 12)
        Update transition priority using Equation
          (8)
    Update the value function networks by
      optimizing the objective (Equation 14):
      $\omega_j = \omega_j - \beta \nabla_{\omega_j} J(\omega_j)$
    if $t\%d == 0$ then
        Update the policy network using the gradient
          (Equation 13):
          $\theta = \theta + \alpha \frac{1}{N} \sum_{i=1}^N [\rho_i \cdot \nabla_\theta u(s_t^{(i)}; \theta) \nabla_a Q(s_t^{(i)}, u(s_t^{(i)}; \theta); \omega_1)]$
        Update the target networks:
        $\omega_j^- = \tau \omega_j + (1 - \tau) \omega_j^-$
        $\theta^- = \tau \theta + (1 - \tau) \theta^-$
return $\theta$
```

Algorithm 1. Training the deterministic continuous VP policy network.

3.5. Training the policy network

In this section, we resort twin delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018) and the scheme designed in Section 3.4 to develop a training algorithm for the solution of the optimal VP policy parameters θ^*. To this end, we use the gradient (Equation 4) to iteratively update θ: $\theta = \theta + \alpha \nabla_\theta J(\theta)$. α is the learning rate. The core task is to solve the gradient $\nabla_\theta J(\theta)$. We therefore employ Monte Carlo method to replace the expected operator in Equation (4) in an approximate manner. Specifically, we sample N transitions from the viewpoint transition buffer using Equation (7) to calculate

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} [\rho_i \cdot \nabla_\theta u(s_t^{(i)}; \theta) \nabla_a Q^u(s_t^{(i)}, u(s_t^{(i)}; \theta))]. \tag{13}$$

According to TD3, we approximately represent the value function $Q^u(s_t, a_t)$ in Equation (13) by a three-layer fully-connected neural network $Q(s_t, a_t; \omega)$ with the parameters ω. The network takes the state s_t and the action a_t as input and outputs the function value $Q(s_t, a_t; \omega)$. By updating the parameters ω, the value function corresponding to the VP policy u can be obtained.

In order to better train the policy network $u(s_t; \theta)$, we follow TD3 to build six neural networks in total: policy network $u(s_t; \theta)$, value function network 1 $Q(s_t, a_t; \omega_1)$, value function network 2 $Q(s_t, a_t; \omega_2)$ and their corresponding target networks [target policy network $u(s_t; \theta^-)$, target value function network 1 $Q(s_t, a_t; \omega_1^-)$, target value function network 2 $Q(s_t, a_t; \omega_2^-)$]. After the training, the policy network $u(s_t; \theta)$ is the optimal deterministic continuous VP policy we want. The other networks only serve as auxiliary training. Figure 3 shows the relationship between the six networks.

The value function networks can be updated with the aforementioned N samples by minimizing the objective

$$L(\omega_j) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_t^{corr(i)} - Q(s_t^{(i)}, a_t^{(i)}; \omega_j))^2 \tag{14}$$

where j is 1 or 2. $\hat{y}_t^{corr}$ is the corrected TD target proposed in Equation (12).

Our whole algorithm to train the deterministic continuous VP policy network is summarized in Algorithm 1. Once the optimal parameters θ^* are obtained after the training, we can use them for the practical AOR task. Given a state s_t, the planned action is $a_t^* = u(s_t; \theta^*)$, and the next best viewpoint of the agent is $\Phi_{t+1} = \Phi_t + a_t^*$.

4. Experiments

This section first provides details about the experimental dataset and implementation, and then reports the experimental results along with some analyzes.

4.1. Dataset and metric

We evaluate our proposed deterministic continuous VP method on the public and challenging dataset GERMS (Malmir et al., 2015) shown in Figure 4A which is collected in the context of developing robots to interact with toddlers in early childhood education environments. The dataset has 1,365 video tracks of give-and-take trials using 136 different object instances. The object instances are soft toys denoting a wide range of disease-related organisms, microbes and human cell types. Each video track records a robot grasping an object instance to its center of view, rotating the object by 180° with its left or right arm, and then returning it. All video tracks were recorded by a head-mounted camera of the robot at 30 frames/s, as shown in Figure 4B. At the same time, the joint position and object label corresponding to each frame image were also recorded in each track. These joint positions provide an opportunity for verifying different VP methods in one dimensional action space. The dataset authors specified the image subsets of all tracks as train and test set, as shown in Table 1. The evaluation metric used for different VP methods is recognition accuracy that is the average value of the entire test set. The higher the recognition accuracy is, the better the corresponding VP method will be.

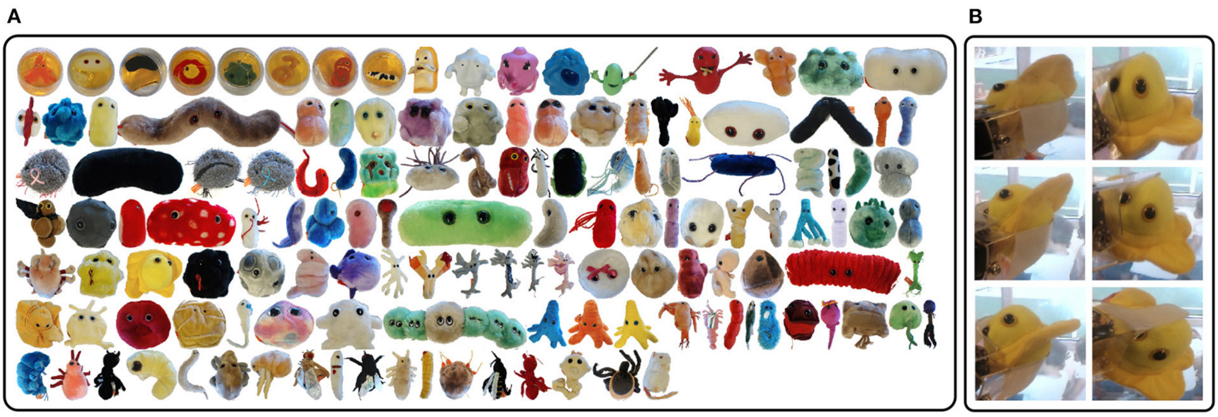

FIGURE 4
The GERMS dataset. **(A)** One hundred and thirty six object instances. **(B)** Recorded images of different joint positions in each track.

4.2. Implementation details

4.2.1. Network architecture

The Tensorflow platform is used to implement the proposed method in this work. In the pre-trained classifier, we transform every image in the GERMS dataset into a 4,096-dimensional feature vector using an existing CNN model VGG-net provided by Malmir et al. (2015). The *softmax* layer has 136 neurons. For the policy network $u(s_t; \theta)$, the dimensions of each layer are 136, 512, 512 and 1. The activation functions of the two hidden layers are both *relu*. The output layer adopts *tanh* activation function, which is multiplied by 512 so as to make the planned relative VP action in $[-45^\circ, 45^\circ]$. For the two value function networks ($Q(s_t, a_t; \omega_1)$ and $Q(s_t, a_t; \omega_2)$), they have the same network structure with the dimensions of each layer are 137, 512, 512 and 1. The activation functions of the two hidden layers are also *relu*. The configuration of their corresponding target network is completely consistent with theirs.

4.2.2. Viewpoint transition management

The capacity of the viewpoint transition buffer is 10^6. ϵ and the exponent λ are set as 0.01 and 0.6 according to the original setting of PER (Schaul et al., 2016). To efficiently sample from distribution (Equation 7), we use a "sum-tree" (Schaul et al., 2016) in which every node is the sum of its children and the leaf nodes are priorities. The sum-tree can be efficiently updated and sampled from.

TABLE 1 GERMS dataset statistics (mean ± std).

Images/track	Number of tracks	Images/track	Total number of images
Train	816	157 ± 12	76,722
Test	549	145 ± 19	51,561

4.2.3. Training

The reward discount factor γ is 0.95. The minibatch size N is 128. The maximum step T_{max} for recognition is $T_{max} = 12$ and the preset probability threshold is 0.99. The Adam optimizer (Kingma and Ba, 2014) is utilized to optimize the policy network and the value function networks. The learning rates are 0.0001, 0.001, and 0.001, respectively. The standard deviations (σ_1 and σ_2) of the exploration noise and smoothing regularization are 128 and 32. c is 512. The delayed update cycle d and soft update τ are 2 and 0.01.

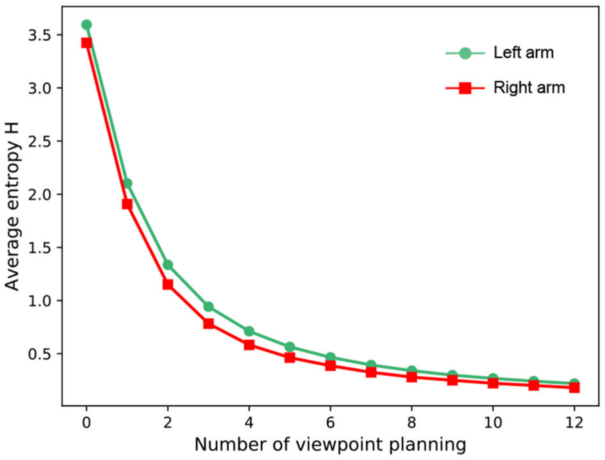

FIGURE 6
The average entropy over the whole test dataset. The experiment is implemented with our VP model.

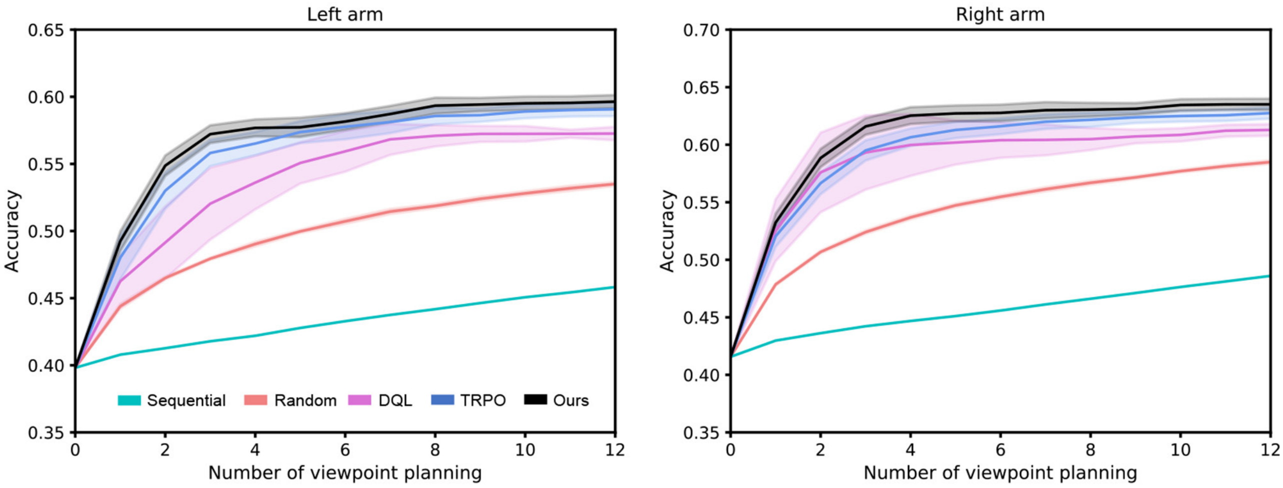

FIGURE 5
Performance comparison between our presented deterministic continuous VP approach and several competing methods. The shaded region represents the standard deviation of the average evaluation over 10 trials.

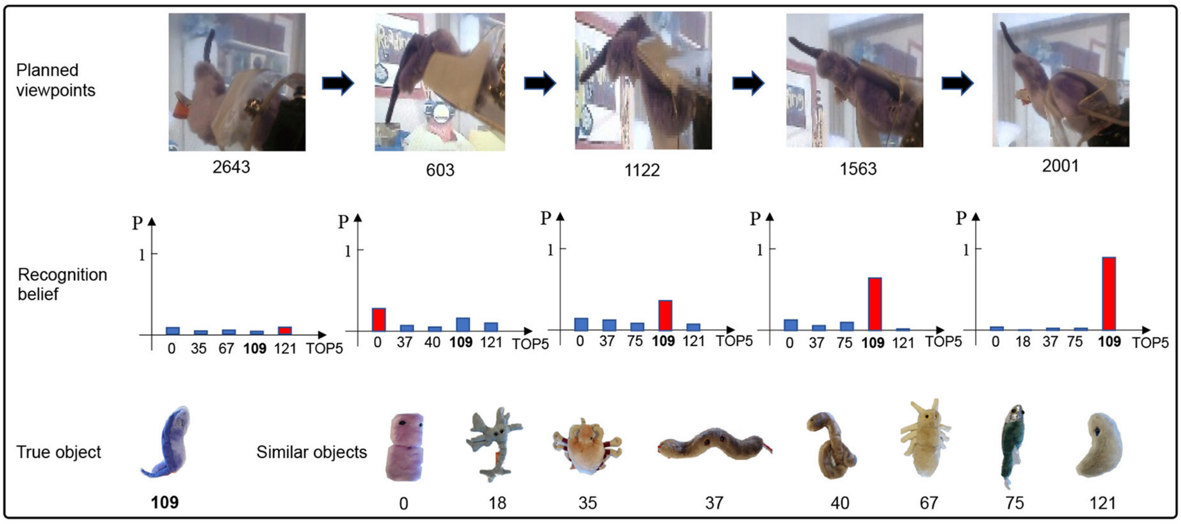

FIGURE 7
An example of actively identifying an object by our VP method. The recognition belief increases with the increase of the number of viewpoint planning.

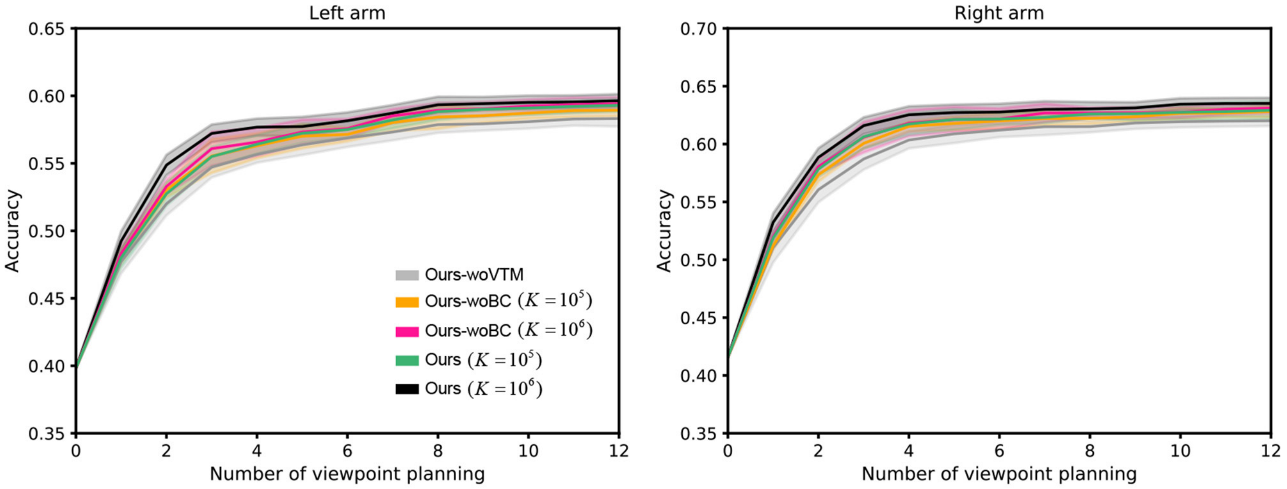

FIGURE 8
The performance comparison results of ablation experiments. *K* represents the capacity of the viewpoint transition buffer. The shaded region represents the standard deviation of the average evaluation over 10 trials.

4.3. Results and analyzes

4.3.1. Comparison with competing methods

To validate the effectiveness of our proposed deterministic continuous VP method in this experiment, we compare our proposed method with the following baseline and competing methods.

4.3.1.1. Single viewpoint recognition

Single viewpoint recognition only allows the agent to recognize an object from one viewpoint.

4.3.1.2. Blind VP policies

Random policy (Liu et al., 2018a) randomly selects an action from the continuous action space $[-45°, 45°]$ with a uniform probability. Sequential policy (Liu et al., 2018a) moves the agent to the next adjacent viewpoint in the same direction. The reason why these two baseline policies are called blind VP policies is that they do not use the previous observation information for purposeful viewpoint planning. The blind policies may produce worthless viewpoints for recognition.

4.3.1.3. Purposeful discrete VP policy

DQL policy (Malmir et al., 2015; Malmir and Cottrell, 2017) develops an active discrete VP method with deep Q-Learning algorithm, which explores in the discrete action space $\{\pm\frac{\pi}{64}, \pm\frac{\pi}{32}, \pm\frac{\pi}{16}, \pm\frac{\pi}{8}, \pm\frac{\pi}{4}\}$.

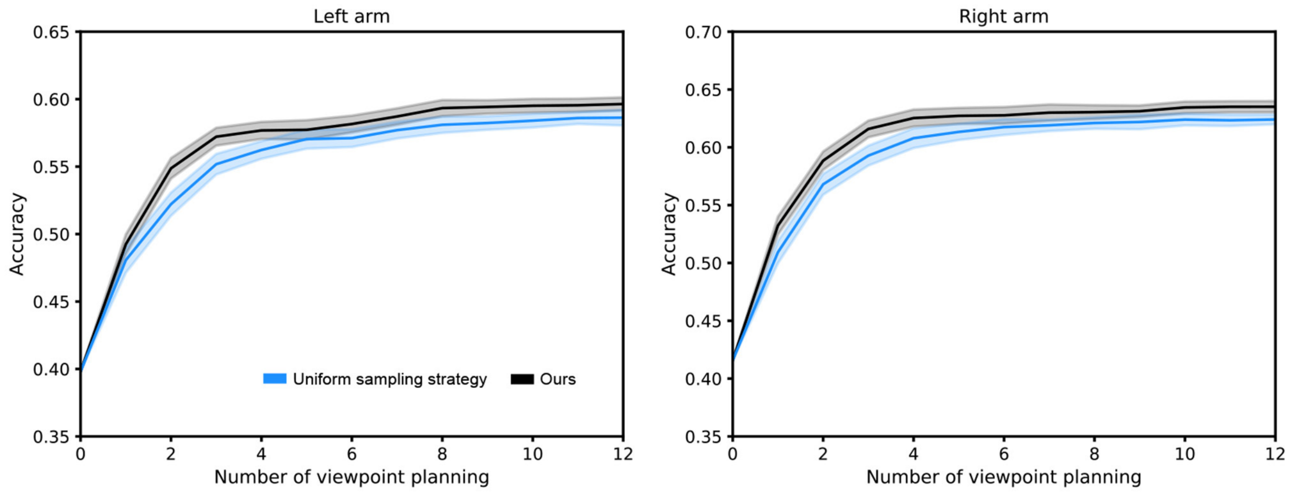

FIGURE 9
Performance comparison between our sampling strategy and uniform sampling strategy. The capacity of the viewpoint transition buffer is 10^6. The shaded region represents the standard deviation of the average evaluation over 10 trials.

4.3.1.4. Purposeful continuous VP policy

TRPO policy (Liu et al., 2018b) utilizes trust region policy optimization (Schulman et al., 2015) to learn a continuous VP policy and adopts extreme learning machine (Huang et al., 2006) to reduce computational complexity. This policy has on-policy characteristic that means the agent can not reuse learned viewpoint transitions for efficient training.

Since the main focus of this work is viewpoint planning, we do not investigate the impact of classifiers on recognition performance. Therefore, for a fair comparison, the classifiers in different approaches are the same in the experiment. Figure 5 reports the experimental results of our method against other approaches over 10 random seeds of the policy network initialization. Some observations from Figure 5 are presented as follows: (1) Viewpoint planning can greatly improve recognition performance. The number of VP is 0 that means the agent recognizes the concerned object with a single viewpoint. Obviously, the recognition accuracy of single viewpoint recognition policy is far lower than that of the methods which perform multi viewpoint recognition *via* VP. This is because more object information with difference can be found through VP to reduce recognition uncertainty, thus improving the recognition performance. As shown in Figure 6, the uncertainty of recognition decreases as the number of viewpoints increases. Figure 7 shows the process of actively identifying an object. (2) The performance of the blind VP policies is nowhere near as good as that of the purposeful VP policies. The primary reason is that the purposeful VP policies (i.e., DQL policy, TRPO policy and our policy) can purposefully plan next viewpoints according to the observed information. (3) The continuous VP policies have better performance than the discrete VP policy. That is because the continuous VP policies (i.e., TRPO policy and our policy) directly explore continuous viewpoint space without sampling, so they will not miss some important viewpoints. (4) The performance of our deterministic continuous VP policy exceeds that of TRPO policy. This is mainly because we design a scheme of viewpoint transition management that can reuse the obtained viewpoint transitions to improve the training effect.

4.3.2. Ablation studies

To verify the importance of different components in our proposed VP model, we intend to conduct the variant experiments with the ablation of different components, i.e., viewpoint transition management (VTM) and bias correction (BC). Training the model without VTM and BC are respectively denoted as Ours-woVTM and Ours-woBC. From the presented results over 10 random seeds in Figure 8, we can notice that: (1) The performance of Ours-woVTM is the worst. It illustrates that our designed scheme of viewpoint transition management indeed enhances the training effect. (2) The performance of Ours-woBC is inferior to that of Ours, especially when the capacity K of the viewpoint transition buffer is large. This is because when the capacity is larger, the distribution of s_{t+1} in the buffer is closer to its true distribution. In this case, the effect of our bias correction based on importance sampling will be more obvious.

4.3.3. Sampling strategies investigations

To verify the superiority of our proposed sampling strategy (i.e., prioritized experience replay based on clipped double Q-learning and bias correction) in the scheme of viewpoint transition management, we conduct comparison experiments with the uniform sampling strategy (Lin, 1992) over 10 random seeds. As shown in Figure 9, we observe that our sampling strategy achieves a better performance, since the importance of each viewpoint transition is ignored by the uniform sampling strategy.

5. Conclusions

In this paper, a continuous viewpoint planning method with transition management is proposed for active object recognition based on reinforcement learning. Specifically, we employ deterministic policy gradient theory to build a learning framework of the viewpoint planning policy. We also design a scheme of viewpoint transition management that can store and reuse the obtained transitions. We develop an algorithm based on twin delayed deep deterministic gradient and the designed scheme to train the policy. Experiments on a public dataset demonstrate the effectiveness of our method. In the future, we will integrate the calibrated probabilistic classifiers in AOR research. As stated in Popordanoska et al. (2022), the way the posterior probability distribution is defined in our work assumes that the classifier is properly calibrated, i.e. the *softmax* output represents the correct error rate probabilities. In general, this is not necessarily the case.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

HS and YL: conceptualization. FZ and HS: methodology. HS and YK: software. HS and SF: investigation. FZ: resources and funding acquisition. YL: data curation. HS: writing—original draft. YL and PZ: writing—review and editing. JW and YW: supervision. All authors have read and agreed to the published version of the manuscript.

Funding

This work was supported by the National Natural Science Foundation of China under Grant No. U1713216.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Andreopoulos, A., and Tsotsos, J. K. (2013). 50 years of object recognition: directions forward. *Comput. Vis. Image Understand.* 117, 827–891. doi: 10.1016/j.cviu.2013.04.005

Becerra, I., Valentin-Coronado, L. M., Murrieta-Cid, R., and Latombe, J.-C. (2014). "Appearance-based motion strategies for object detection," in *2014 IEEE International Conference on Robotics and Automation (ICRA)* (Hong Kong: IEEE), 6455–6461.

Behl, A., Hosseini Jafari, O., Karthik Mustikovela, S., Abu Alhaija, H., Rother, C., and Geiger, A. (2017). "Bounding boxes, segmentations and object coordinates: how important is recognition for 3d scene flow estimation in autonomous driving scenarios," in *Proceedings of the IEEE International Conference on Computer Vision* (Venice: IEEE), 2574–2583.

Bellemare, M. G., Dabney, W., and Munos, R. (2017). "A distributional perspective on reinforcement learning," in *International Conference on Machine Learning* (PMLR), 449–458. doi: 10.48550/arXiv.1707.06887

Duan, J., Bello, G., Schlemper, J., Bai, W., Dawes, T., Biffi, C., et al. (2019). Automatic 3D bi-ventricular segmentation of cardiac images by a shape-refined multi-task deep learning approach. *IEEE Trans.* 38, 2151–2164. doi: 10.1109/TMI.2019.2894322

Fujimoto, S., Hoof, H., and Meger, D. (2018). "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning* (PMLR), 1587–1596. doi: 10.48550/arXiv.1802.09477

Hao, G., Fu, Z., Feng, X., Gong, Z., Chen, P., Wang, D., et al. (2021). A deep deterministic policy gradient approach for vehicle speed tracking control with a robotic driver. *IEEE Trans. Autom. Sci. Eng.* 19, 2514–2525. doi: 10.1109/TASE.2021.3088004

Huang, G.-B., Zhu, Q.-Y., and Siew, C.-K. (2006). Extreme learning machine: theory and applications. *Neurocomputing* 70, 489–501. doi: 10.1016/j.neucom.2005.12.126

Jayaraman, D., and Grauman, K. (2019). End-to-end policy learning for active visual categorization. *IEEE T. Pattern Anal.* 41, 1601–1614. doi: 10.1109/TPAMI.2018.2840991

Kingma, D. P., and Ba, J. (2014). ADAM: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980.* doi: 10.48550/arXiv.1412.6980

Liang, Y., Guo, C., Ding, Z., and Hua, H. (2020). Agent-based modeling in electricity market using deep deterministic policy gradient algorithm. *IEEE Trans. Power Syst.* 35, 4180–4192. doi: 10.1109/TPWRS.2020.2999536

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., et al. (2015). Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971.*

Lin, L.-J. (1992). *Reinforcement Learning for Robots Using Neural Networks.* Pittsburgh, PA: Carnegie Mellon University.

Liu, H., Li, F., Xu, X., and Sun, F. (2018a). Active object recognition using hierarchical local-receptive-field-based extreme learning machine. *Memet. Comput.* 10, 233–241. doi: 10.1007/s12293-017-0229-2

Liu, H., Wu, Y., and Sun, F. (2018b). Extreme trust region policy optimization for active object recognition. *IEEE Trans. Neural Network Learn. Syst.* 29, 2253–2258. doi: 10.1109/TNNLS.2017.2785233

Malmir, M., and Cottrell, G. W. (2017). "Belief tree search for active object recognition," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE), 4276–4283.

Malmir, M., Sikka, K., Forster, D., Movellan, J. R., and Cottrell, G. (2015). "Deep q-learning for active recognition of germs: Baseline performance on a standardized dataset for active learning," in *Proceedings of the British Machine Vision Conference (BMVC)*, 161.1–161.11. doi: 10.5244/C.29.161

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., et al. (2013). Playing atari with deep reinforcement learning. *arXiv preprint.* doi: 10.48550/arXiv.1312.5602

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., et al. (2015). Human-level control through deep reinforcement learning. *Nature* 518, 529–533. doi: 10.1038/nature14236

Paletta, L., and Pinz, A. (2000). Active object recognition by view integration and reinforcement learning. *Robot. Auton. Syst.* 31, 71–86. doi: 10.1016/S0921-8890(99)00079-2

Parr, T., Sajid, N., Da Costa, L., Mirza, M. B., and Friston, K. J. (2021). Generative models for active vision. *Front. Neurorobot.* 15, 651432. doi: 10.3389/fnbot.2021.651432

Patten, T., Zillich, M., Fitch, R., Vincze, M., and Sukkarieh, S. (2015). Viewpoint evaluation for online 3-d active object classification. *IEEE Robot. Autom. Lett*. 1, 73–81. doi: 10.1109/LRA.2015.2506901

Popordanoska, T., Sayer, R., and Blaschko, M. B. (2022). A consistent and differentiable lp canonical calibration error estimator. *arXiv preprint*. doi: 10.48550/arXiv.2210.07810

Potthast, C., Breitenmoser, A., Sha, F., and Sukhatme, G. S. (2016). Active multi-view object recognition: a unifying view on online feature selection and view planning. *Robot Auton. Syst*. 84, 31–47. doi: 10.1016/j.robot.2016.06.013

Roynard, X., Deschaud, J.-E., and Goulette, F. (2018). "Paris-lille-3d: a point cloud dataset for urban scene segmentation and classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops(CVPR)* (Salt Lake City, UT: IEEE), 2027–2030.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2016). "Prioritized experience replay," in *Proceedings of the International Conference on Learning Representations (ICLR)*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015). "Trust region policy optimization," in *International Conference on Machine Learning* (PMLR), 1889–1897. doi: 10.48550/arXiv.1502.05477

Shi, Q., Lam, H.-K., Xuan, C., and Chen, M. (2020). Adaptive neuro-fuzzy pid controller based on twin delayed deep deterministic policy gradient algorithm. *Neurocomputing* 402, 183–194. doi: 10.1016/j.neucom.2020.03.063

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., and Riedmiller, M. (2014). "Deterministic policy gradient algorithms," in *International Conference on Machine Learning* (PMLR), 387–395.

Stria, J., and Hlavác, V. (2018). "Classification of hanging garments using learned features extracted from 3d point clouds," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Madrid: IEEE), 5307–5312.

Sun, Y., Ran, X., Zhang, G., Wang, X., and Xu, H. (2020). Auv path following controlled by modified deep deterministic policy gradient. *Ocean Eng*. 210, 107360. doi: 10.1016/j.oceaneng.2020.107360

Sutton, R. S., and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. Cambridge: MIT Press.

Van de Maele, T., Verbelen, T., Çatal, O., and Dhoedt, B. (2022). Embodied object representation learning and recognition. *Front. Neurorobot*. 16, 840658. doi: 10.3389/fnbot.2022.840658

Wang, Y., Sun, J., He, H., and Sun, C. (2020). Deterministic policy gradient with integral compensator for robust quadrotor control. *IEEE Trans. Syst. Man Cybernet. Syst*. 50, 3713–3725. doi: 10.1109/TSMC.2018.2884725

Wu, J., Yang, Z., Liao, L., He, N., Wang, Z., and Wang, C. (2022). A state-compensated deep deterministic policy gradient algorithm for uav trajectory tracking. *Machines* 10, 496. doi: 10.3390/machines10070496

Wu, K., Ranasinghe, R., and Dissanayake, G. (2015). "Active recognition and pose estimation of household objects in clutter," in *2015 IEEE International Conference on Robotics and Automation (ICRA)* (Seattle, WA: IEEE), 4230–4237.

Zeng, R., Wen, Y., Zhao, W., and Liu, Y.-J. (2020). View planning in robot active vision: a survey of systems, algorithms, and applications. *Comput. Vis. Media* 6, 225–245. doi: 10.1007/s41095-020-0179-3

Zhang, T., Zhu, K., and Wang, J. (2020). Energy-efficient mode selection and resource allocation for d2d-enabled heterogeneous networks: a deep reinforcement learning approach. *IEEE T. Wirel. Commun*. 20, 1175–1187. doi: 10.1109/TWC.2020.3031436

Zhao, D., Chen, Y., and Lv, L. (2016). Deep reinforcement learning with visual attention for vehicle classification. *IEEE Tran. Cogn. Dev. Syst*. 9, 356–367. doi: 10.1109/TCDS.2016.2614675