





# Traded Control of Human–Machine Systems for Sequential Decision-Making Based on Reinforcement Learning

Qianqian Zhang , Yu Kang , Senior Member, IEEE, Yun-Bo Zhao , Senior Member, IEEE, Pengfei Li , and Shiyi You

**Abstract**—Sequential decision-making (SDM) is a common type of decision-making problem with sequential and multistage characteristics. Among them, the learning and updating of policy are the main challenges in solving SDM problems. Unlike previous machine autonomy driven by artificial intelligence alone, we improve the control performance of SDM tasks by combining human intelligence and machine intelligence. Specifically, this article presents a paradigm of a human–machine traded control systems based on reinforcement learning methods to optimize the solution process of sequential decision problems. By designing the idea of autonomous boundary and credibility assessment, we enable humans and machines at the decision-making level of the systems to collaborate more effectively. And the arbitration in the human–machine traded control systems introduces the Bayesian neural network and the dropout mechanism to consider the uncertainty and security constraints. Finally, experiments involving machine traded control, human traded control were implemented. The preliminary experimental results of this article show that our traded control method improves decision-making performance and verifies the effectiveness for SDM problems.

**Impact Statement**—The human–machine SDM problem refers to the SDM problem in which humans and machines participate together. They alleviate the burden on human decision-makers and also allow humans to have more final decision-making authority than fully autonomous machine control. At present, there are very few methods to study the problem of human–machine SDM, and the human–machine collaboration involved in this field lacks powerful exploration. The traded control method of the human–machine systems proposed by us provides a solution to the human–machine SDM. It can provide help for the field of human–machine co-driving, human–machine minimally invasive surgery, and other more decision-making problems involving humans and machines.

**Index Terms**—Human–machine systems, reinforcement learning, sequential decision-making (SDM), traded control.

Manuscript received July 14, 2021; revised September 18, 2021; accepted November 7, 2021. Date of publication November 12, 2021; date of current version July 21, 2022. This work was supported by the National Key Research and Development Program of China under Grant 2018AAA0100801. This paper was recommended for publication by Associate Editor Douglas S. Lange upon evaluation of the reviewers' comments. (*Corresponding author: Yu Kang.*)

Qianqian Zhang, Yun-Bo Zhao, Pengfei Li, and Shiyi You are with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: zqq789@mail.ustc.edu.cn; ybzhao@ieee.org; puffylee@mail.ustc.edu.cn; ysy3765@mail.ustc.edu.cn).

Yu Kang is with the Department of Automation, Institute of Advanced Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: kangduyu@ustc.edu.cn).

Digital Object Identifier 10.1109/TAI.2021.3127857

## I. INTRODUCTION

SEQUENTIAL decision-making (SDM) problems marked by sequentiality and multistage are a kind of important decision-making problems that widely exist in various fields such as society, economy, military, and industrial production. Because the decision space of this kind of decision-making problem increases exponentially with the decision step length, it is often difficult to obtain the optimal decision sequence. It is worth mentioning that the rapid development of artificial intelligence (AI) [1]–[4] in recent years has also continuously promoted the development of SDM. SDM has as a consequence the intertemporal choice problem, where earlier decisions influence the later available choices. In some practical projects, after making a decision, some new situations arise, and new decisions need to be made. For example, robot motion planning [5], [6], assisted driving systems [7], [8], and minimally invasive surgery [9]–[11] can all be modeled as different SDM problems. Therefore, the study of the sequential decision itself is helpful to promote the development of related engineering fields.

The main challenge in solving the SDM problem is that policy updates cannot keep up with changes in the dynamic environment. On the one hand, for fully autonomous machines driven by AI, their uncertainty, credibility, and vulnerability to attacks cannot be ignored. When faced with a dynamic, uncertain, and changing environment, the consequences of not making timely decisions are serious, even related to life. For example, self-driving cars in an open environment are difficult to use with peace of mind due to the unknown and uncertain environment. After all, once there is a slight difference in perception or decision-making, the risk will be fatal and unrecoverable. On the other hand, although individual human operators can meet certain intentions and cognitive needs, their low-precision operations, and labor costs make them unable to be in an advantageous position in most industrialization and informatization processes. For example, humans cannot perform minor operations on their own, and enter dangerous environments to perform operations (various search and rescue operations), etc.

In response to the above-mentioned challenges, there have been many related studies at this stage to improve it to varying degrees based on various methods. Alagoz [10] constructed and explained the markov decision processes (MDP) model, a powerful analysis tool for SDM under uncertainty, and introduced its

application in the field of medical decision-making (MDM). To analyze the complexity of verifying the correctness of deep neural network (DNN) and a lack of safety guarantees, Michelmore *et al.* [12] extracted a quantitative measure of uncertainty based on Gal's conclusions [13] that dropout neural network could approximate Bayesian neural network (BNN), and they evaluated its quality in the end-to-end controller of the self-driving car. In addition, the method of solving SDM problems based on deep reinforcement learning has been developed rapidly in the past five years. The most representative deep reinforcement learning algorithms are the deep Q network (DQN) proposed by Mnih [14], as well as the subsequent deep deterministic policy gradient (DDPG) [15], asynchronous advantage actor-critic (A3C) [16], and rainbow [17]. For machine learning algorithms, including the DNN and DRL mentioned above, since there is no clear training direction in the process of exploring the optimal policy, a large number of training times are required to achieve the effect that the user wants, and the generalization ability of the policy is weak for data beyond the training set.

Different from the above methods, this article considers the mechanism of human-on-the-loop (HOTL) (its key word involves "traded control") to accelerate the policy learning process and then improve the systems' control performance. The rapid development of machine intelligence is certainly beneficial to real-life engineering, but the machine intelligence that has developed so far cannot completely replace human intelligence. Therefore, considering the integration of machine intelligence and human intelligence is a new idea to solve the challenges described in this article. Unlike classical machines that implement SDM completely autonomously [14], [18]–[20], this article considers that human–machine traded control systems implement SDM, wherein the several partners need to cooperate to complete the task. Importantly, we emphasize the determination and use of the autonomous boundary, which has not been mentioned in previous traded control [21]–[23], and we will describe it in detail in Section III. Human–machine traded control in this article includes two situations: the scene where the machine traded with the human (we call it machine traded control for short); and the scene where the human traded with the machine (we call it human traded control for short). We briefly list the contribution points as follows.

- 1) We put forward the concept of autonomous boundary, and gave its definition and judgment method. The existence of the autonomous boundary make the decision-making authority of human and machine in the traded control systems have a preliminary clear division.
- 2) Based on the judgment of the autonomous boundary, we optimize the design of the human–machine traded control systems, which is conducive to improving the overall decision-making performance of the system.
- 3) We divide the scene of human–machine traded control systems into machine traded control and human traded control. And we optimize the design for these two types of scenarios to meet the application requirements of more different control subjects.
- 4) Taking LunarLander as a representative example, we verified the method proposed in this article and obtained good

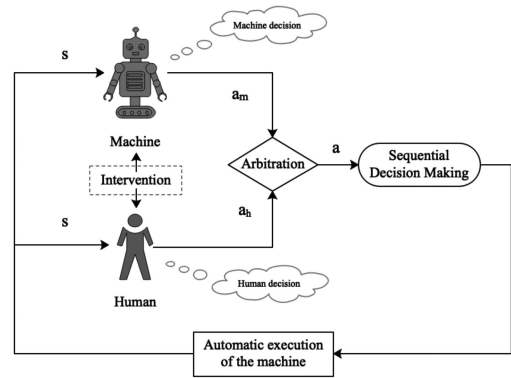


Fig. 1. Human–machine traded control framework for SDM.

preliminary experimental results. This has a positive role in promoting and encouraging more relevant researchers to continue to explore the solution of human–machine SDM problems.

The rest of this article is organized as follows. Section II introduces problem statement and related work, background knowledge, respectively. Section III describes the proposed framework and main methods in detail and the experimental results are shown in Section IV. Finally, Section V concludes this article.

## II. PROBLEM STATEMENT AND RELATED WORK

Our work takes MDP (see [24], [25]) as the theoretical background to study the application of human–machine traded control systems in the field of sequential decision-making.

### A. Problem Statement

The fact that human–machine performance is better than purely human performance should come as no surprise for the situations, where machines outperform humans are usually standard, routined, automated, repeated, noncreative. Additionally, humans' inherent cognitive advantages enable them to make instructive decision-making behaviors until the machine agent learns a good strategy. When the environment is suddenly changed or is subject to huge disturbances, the machine agent may not be able to meet the decision-making needs in a timely manner, e.g., fierce military human-UAV combat. Therefore, considering the respective advantages of humans and machines, human intelligence cannot be ignored before the machine achieves complete intelligent autonomy in various fields. The question then naturally arise for us to integrate human and machine intelligence, for better, augmented intelligence, as have been investigated very briefly in our present work.

As mentioned in Section I, we try to obtain the solution of sequential decision-making through traded control of HOTL, as shown in Fig. 1. Fig. 1 shows the general framework of the human–machine traded control systems (including the above two situations: machine traded control, human traded control). To avoid redundancy, we put human traded control in parentheses and describe it at the same time. For machine traded control (human traded control), human (machine) plays the role

of regular decision-makers. The machine agent (human) is not so much in the control loop as it is on the control loop. Therefore, the machine (human) plays the role of supervisor or substitute. When a serious error occurs in the decision-making behavior of the human (machine), the machine (human) can forcefully intervene in the decision-making until the human (machine) resumes normal decision-making ability.

Regardless of the above two situations, how to trigger the occurrence of trade and the degree of triggering are both important and difficult. Facing SDM, we discuss the autonomous boundary problem, decision uncertainty, and reasonable arbitration mechanism in the human–machine traded control systems based on reinforcement learning to achieve the improvement of decision-making performance. But in order to describe these problems uniformly, we do not emphasize the direction of trade here for the time being. Human intelligence and machine intelligence are combined in an “traded control” way to improve decision-making performance. We describe an optimization model as follows:

$$\max_{a_t \in A} J(s(t), a(t)) = Q(s(t), a(t)) \quad (1a)$$

$$\text{s.t. } a(t) = f^a(s(t), a_m(t), a_h(t), b(t)) \quad (1b)$$

$$a_m(t) = p_m(s(t)) \quad (1c)$$

$$a_h(t) = \text{Human} - \text{Action} \quad (1d)$$

$$s(t+1) = f^d(s(t), a(t)) \quad (1e)$$

$$\begin{aligned} C(s(t), a_m(t), a_h(t)) &\leq 0 \\ t &= 0, 1, 2, 3, \dots \end{aligned} \quad (1f)$$

where  $f^a(\cdot)$  determines which decision-maker corresponds to the current state of the environment.  $a_m(t), a_h(t)$  represent the action of machine and human, respectively.  $a(t)$  describes the result of the arbiter and also represents the action that the system should take at time  $t$ .  $b(t)$  is autonomous boundary.  $Q(s(t), a(t))$  is the value function of the state-action pair  $(s(t), a(t))$ .  $s(t), s(t+1)$  describes the state at time  $t$  and  $t+1$ , respectively.  $f^d(\cdot)$  represents the state transition model.  $C(\cdot)$  constrains states and actions.

In this article, we combine machine intelligence and human intelligence to achieve improvements to fully autonomous machine algorithms. How to establish a unified human–machine traded control systems framework, which can affect the next time step state information through a higher-quality decision method under the given environmental state observation (such as position information and attitude information), is our concern. We will introduce the specific methods in detail in Section III.

## B. Related Work

1) *Human–Machine Traded Control*: Traded control is based on a certain evaluation mechanism or task objective, and the strong decision makers’ compulsory control over the weak decision makers. This article mainly focuses on the human–machine traded control systems. This method provides an alternative to fully automated robotic systems that can be used to expand the effectiveness of modern robots in more user-friendly areas, such

as assisted driving, autonomous weapon systems, and intelligent learning assistance systems.

In traded control, the machine or human agent has exclusive control of a system at any point in time. Mixed-initiative trades in control can be proposed by either the machine or human based on agent-specific models of failure probability. Disagreement stems from differences in the agents’ models of failure, and occurs when the agents do not agree to a proposed trade [21]. According to [22], in traded control the operator and the robot both controlled the robot’s actions. The operator initiated a task or behavior for the robot. The robot then performed the task autonomously by following the desired input while the operator monitors the robot. Muir [23] developed a model of human trust in machines, taking models of trust between people as a starting point, and extending them to the human–machine relationship, so as to assist in the completion of human intervention in automated systems. Lex [26] proposed an arguing machine framework based on the idea of integration that the primary and secondary system solve the same control task at the same time. When the two subsystems make relatively different decisions, humans acting as supervisors unilaterally intervene in intelligent machines. In contrast, the literature [27] relied on data-driven, joint human–machine systems to model-based representations to evaluate a large number of potential inputs that users may wish to provide in parallel. This way enabled users to do whatever they want (maximum permissions assigned to human partners) in situations where it is difficult to obtain or without user goals and improves systems’ security.

2) *Arbitration*: Arbitration is necessary for switching or mixing between human and intelligent machines. Arbitration refers to a fusion policy. A common form of fusion in human–machine shared control systems is through a linear combination between the user and autoagent policies. The arbitration parameter may depend on different factors, such as the confidence in the user’s intention prediction, or the difference between each command [6], [28], [29]. The literature [30] evaluated the confidence of the DQN through the network loss function and assessed the degree of agreement between human action recommendations and the actions selected by the DQN. Then, the arbitrator will choose actions between random exploration, the action chosen by the DQN, and human action suggestions (if any). Learning arbitration policies from user interaction was described (this is done by calculating the best fusion parameters afterward, and supervised learning to train the recurrent neural network (RNN) to predict the best arbitration) [28].

3) *Reinforcement Learning*: Reinforcement learning, as an effective method to realize human–machine traded control, has been used in a large number of studies in recent years. [31] proposed a human–machine framework based on model-free reinforcement learning, which took observations of the environment and user’s control or inferred targets (if available) as inputs, and produced high-value actions or control outputs as close as possible to user control. [30] modeled the confidence and consistency of human feedback by extending deep reinforcement learning, thereby using discrete human feedback to enhance the performance of deep learning agents in a virtual 3-D environment (Minecraft). [32] introduced a robot setup that



enables human–robot teams to not only solve collaborative tasks within 30 min in real-world training, but also has the same capabilities as human teams performance. The sampling efficiency of the latest DRL method enables human-in-loop training from scratch, which opens the door for further research on collaborative learning. The author proposed a new human–machine collaborative reinforcement learning algorithm  $CQ(\lambda)$ , which can converge faster than the traditional  $Q(\lambda)$  reinforcement learning algorithm. The algorithm  $CQ(\lambda)$  provided the robot with self awareness to adaptively switch its collaboration level from autonomy to semiautonomy.

4) *Bayesian Neural Network*: Thanks to earlier work [33]–[36], the application of BNN was gradually becoming a reality [12], [37]. What it means by its name was a technique involving Bayesian inference and neural networks. BNN is different from general neural networks in that their weight parameters are random variables with a probability distribution, not definite values. And BNN combines probabilistic modeling with neural networks and gives confidence in the prediction results. *A priori* is used to describe key parameters and is used as input to the neural network. The output of a neural network is used to describe the likelihood of a particular probability distribution. Finally, the posterior distribution is calculated by sampling or variational inference. The BNN is modeled as follows.

Given a dataset  $D = \{X, Y\}$ , training a BNN with the parameter  $\theta$ , we can get the posterior distribution of  $p(\theta|D)$  and the space of functions  $f^\theta$  theoretically. Furthermore, for the new input  $x', y'$  obeys the distribution

$$p(y'|x', D) = \int p(y'|x', \theta)p(\theta|D)d\theta \quad (2)$$

where

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} \quad (3)$$

with  $p(\theta|D)$  posterior distribution,  $p(D|\theta)$  likelihood function,  $p(D)$  marginal likelihood. Since  $\theta$  is a random variable, our predicted value is also a random variable. It can be seen from formula (2) that the core of probabilistic modeling and prediction of data with BNN is to make an efficient approximate posterior inference, and variational inference is a very suitable method. As in the literature [13], we implement approximate inference of BNN based on dropout NN.

### III. METHODOLOGY

For human–machine traded control, how to trigger the occurrence of trade is difficult points that need to be solved urgently.

- 1) The machine traded control means that the control process of the controlled object is originally controlled by humans. When human decision-making errors or extremely uncertain cognition occur, the machines take the initiative to take over the control process of the controlled object.
- 2) Human traded control means that the control process of the controlled object is originally controlled by machines. When the quality of the machines' decisions are extremely poor or the credibility is low, the human partners take the

initiative to take over the control process of the controlled object.

Obviously, it is necessary to judge the decisions of humans and machines, and it is a process of trade from those with high decision-making quality to those with low decision-making quality. In the above two cases, since trade and handover are required in real time, it is particularly important to judge when to trigger the trade and switching. And since humans and machines belong to two control systems with different attributes after all, whether frequent trade will cause the instability of the control process of the controlled object also needs to be considered. Therefore, this article considers how to improve the control performance of the controlled object under traded control, through the effective decision-making behavior generated by modules such as the learned decision network, decision-making quality evaluation method, and fair and reasonable switching mechanism based on the perceived environmental state information.

#### A. Autonomous Boundary of Human–Machine Traded Control

In traded control, when human intelligence and machine intelligence appear at the decision-making level at the same time, developers inevitably have to divide the decision-making authority between humans and machines. Of course, from another perspective, if there is a scope of decision-making authority, there are constraints on decision-making behavior. And we believe that the human behavior space and the machine behavior space constituted by decision-making behaviors that meet this constraint are beneficial to the optimized solution of the controlled object. In this section, we hope to make up for the vacancy of the human–machine traded control systems on this point, using information such as the perceived environmental state and the learned decision-making network to carry out mathematical formal expressions and algorithms for the human–machine decision-making authority in the human–machine traded control systems. Before introducing the autonomous boundaries, we first give a definition of autonomy.

*Definition III.1 (Autonomous boundary of human)*: The autonomous boundary of human refers to the range of human intelligence to make decisions and actions in accordance with the direction that is beneficial to the joint optimization of the human–machine traded control systems.

In general, the autonomous boundary is composed of its lower and upper bounds. However, the lower bound of human autonomy involves the cognitive defects of human beings and is beyond our consideration. Therefore, the question of the upper bound of human autonomy is considered in this article.

In the strategic design of the machine traded control systems, the autonomous boundary of human is an important concept, which relates to when and how the machine intervenes in the human control. When this boundary is not exceeded, the system satisfies the human control decision-making vision. When this boundary is exceeded, the system allows machine trade to occur. With the progress of the dynamic decision-making process, the boundary of human autonomy can be optimized in real time, and the optimized boundary can be used as the next decision condition again. Therefore, we consider defining the boundary

problem of human autonomy as the following optimization problem:

$$b_h(t) = \arg \max_{a_h \in \mathcal{A}_h(t)} J_{h,m}^b(s(t), a_h) \quad (4a)$$

$$\text{s. t. } C(s(t), a_h) < 0 \quad (4b)$$

where

$$\mathcal{A}_h(t) := \{a_h(t), t \geq 0\}. \quad (4c)$$

Among them,  $J_{h,m}^b(s(t), a_h(t))$  is the common objective function of humans and machines. The objective function can be defined as different expressions according to specific implementation scenarios and algorithms, such as cumulative rewards (cost) function

$$J_{h,m}^b(s(t), a_h(t)) = \int_t^{t+T} [r(s(t), a_h(t)) - c(s(t), a_h(t))] dt. \quad (5)$$

*Definition III.2 (Autonomous boundary of machine):* The autonomous boundary of the machine refers to the boundary of the AI-driven machine intelligence to make decisions and actions in the direction that is beneficial to the joint optimization of the human–machine traded control systems.

Similarly, the lower bound of machine autonomy involves the level of mechanical, physical and mechanical automation, and is not at the level of intelligent decision-making that we consider. Therefore, this article considers the upper bound of machine autonomy.

In the strategic design process of human traded control systems, the autonomous boundary of the machine is an important concept, which relates to when and how the human partner intervenes in the machine control. When this boundary is not exceeded, the demand for autonomous control and decision-making of the machine is met, and when this boundary is exceeded, human partner trade occurs. With the progress of the decision-making process, the autonomous boundary of the machine can be optimized in real time, and the optimized boundary can be used as a judgment condition in the future. Therefore, similarly, we consider defining the autonomous boundary problem of the machine as an optimization problem

$$b_m(t) = \arg \max_{a_m \in \mathcal{A}_m(t)} J_{h,m}^b(s(t), a_m) \quad (6a)$$

$$\text{s. t. } C(s(t), a_m) < 0 \quad (6b)$$

where

$$\mathcal{A}_m(t) := \{a_m(t), t \geq 0\}. \quad (6c)$$

Among them,  $J_{h,m}^b(s(t), a_m(t))$  is the common objective function of humans and machines. The definition of  $J_{h,m}^b(s(t), a_m(t))$  is similar to (5).

Considering the optimization problem of the above (4) and (6), we can describe the general algorithm idea as shown in Algorithm 1. First, we initialize the upper bound of human (machine) autonomy, for instance, in a manner similar to the random initialization of parameters in a neural network. In the process of dynamic evolution, the real-time decision-making behavior of humans (machine intelligence) is filtered based on constraints.

---

**Algorithm 1: Algorithm Design of Autonomous Boundary.**


---

- 1: **Initialization:** Autonomous boundary of human  $\bar{B}_h = \{b_h(0)\}$  (machine  $\bar{B}_m = \{b_m(0)\}$ ).
  - 2: **Output:** Autonomous boundary of human  $\bar{B}_h = \{b_h(t)\}$  (machine  $\bar{B}_m = \{b_m(t)\}$ ).
  - 3: **repeat**
  - 4:   Input: The system state  $s(t)$ , human behavior  $a_h(t)$  (machine behavior  $a_m(t)$ ).
  - 5:   Filter the human behavior  $a_h(t)$  (machine behavior  $a_m(t)$ ) at time  $t$  according to the constraints (4b) (or (6b)) in the optimization (4) (or optimization (6)).
  - 6:   Compare the objective function (4a) ((6a)) corresponding to human behavior  $a_h(t)$  (machine behavior  $a_m(t)$ ) with the boundary information  $b_h(t-1)$  (or  $b_m(t-1)$ ). If  $J_{h,m}^b(s(t), a_h(t)) > J_{h,m}^b(s(t), b_h(t-1))$  (or  $J_{h,m}^b(s(t), a_m(t)) > J_{h,m}^b(s(t), b_m(t-1))$ ), the upper bound of human autonomy  $b_h(t)$  (or the upper bound of machine autonomy  $b_m(t)$ ) is calculated according to the optimized expression (4) (or according to the optimized expression (6)).
  - 7: **until** End of training.
- 

After that, by comparing the real-time human decision-making action (machine decision) at time  $t$  with the upper bound of human autonomy  $b_h(t-1)$  (or the upper bound of machine autonomy  $b_m(t-1)$ ) at the previous moment  $t-1$ , based on optimization (4) [or (6)], the autonomy boundary of human  $b_h(t)$  [the upper bound of machine autonomy  $b_m(t)$ ] at the current moment is updated. Taking maximizing the objective function as an example, the behavior corresponding to the maximum objective function within the constraint range is the information of boundary we want to search. So far, we have given a general method to determine the autonomous boundary of humans and machines.

### B. Optimal Design of Human–Machine Traded Control

Traded control means that the status of decision makers in the human–machine systems is asymmetrical. Either it is the human–machine master–slave relationship that meets the needs of human experiences, or the machine–human master–slave relationship that emerges by using the machine’s high-precision capabilities. This section considers the optimization of the design of the human–machine traded control systems based on the above discussion of the autonomous boundary (see Section III-A). Considering that the machine can intervene in human control in one direction is a realistic requirement, which means that the machine has a higher priority decision-making authority. In this case, allowing the mandatory trade of machine intelligence, or even temporarily depriving human autonomy, has become a feasible human–machine collaboration strategy. In the human traded control systems, humans have higher priority decision-making authority. In other words, when the machine controls the operation of the controlled object, if human find



Next, we give an optimization algorithm for traded control based on the autonomous boundary, including the machine traded human control algorithm based on the autonomous boundary (MTHA-B) and the human traded machine based on the autonomous boundary (HTMA-B). Equations (10) and (11) shown at bottom of the page, represent the arbitration function of MTHA-B and HTMA-B, respectively. In machine traded human control, if machine behavior is better than human behavior (including autonomous boundary of human and human real-time input), and the credibility of machine decision-making is greater than human control, then machine's trade triggers success. When the autonomous boundary of human is superior to machine behavior and human real-time input, and the credibility of human decision-making is greater than that of the machine, then the machine's trade is not triggered, but at this time the optimal decision-making behavior is obtained on the autonomous boundary of human. The rest of the cases in machine traded human control indicate that the machine failed to intervene, and the human behavior at this time is the optimal behavior.

In human traded machine control, if human behavior is better than machine behavior (including autonomous boundary of machine and machine real-time input), and human decision-making credibility is greater than machine, then human trade can trigger success. When the autonomous boundary machine is superior to human behavior and machine input, and the decision-making credibility of the machine is greater than that of humans, then human trade is not triggered, but at this time the optimal decision-making behavior is obtained on the autonomous boundary of machine. The rest of the cases in human traded machine control indicate that humans have failed to intervene, and the behavior of the machine at this time is the optimal behavior.

In addition,  $c_m(t)$  and  $c_h(t)$  in (10) and (11) represent the credibility assessment of machine behavior  $a_m(t)$  and human behavior  $a_h(t)$ , respectively. Considering the probabilistic characteristics of the BNN, the arbiter uses the MC dropout [13]

method to measure the credibility of the subsystems decision.  $c_m(t)$  and  $c_h(t)$  can be calculated as follows:

$$\mathbb{E}[a_m(t)] \approx \frac{1}{T} \sum_{i=1}^T p_m^i(s(t)) \quad (12a)$$

$$\mathbb{E}[(a_m(t))^T(a_m(t))] \approx \tau^{-1}I + \frac{1}{T} \sum_{i=1}^T p_m^i(s(t))^T p_m^i(s(t)) \quad (12b)$$

$$c_m(t) = \mathbb{E}[(a_m(t))^T(a_m(t))] - \mathbb{E}[(a_m(t))]^T \mathbb{E}[a_m(t)] \quad (12c)$$

$$\mathbb{E}[a_h(t)] \approx \frac{1}{T} \sum_{i=1}^T a_h^i(t) \quad (13a)$$

$$\mathbb{E}[(a_h(t))^T(a_h(t))] \approx \tau^{-1}I + \frac{1}{T} \sum_{i=1}^T a_h^i(t)^T a_h^i(t) \quad (13b)$$

$$c_h(t) = \mathbb{E}[(a_h(t))^T(a_h(t))] - \mathbb{E}[(a_h(t))]^T \mathbb{E}[a_h(t)] \quad (13c)$$

where  $p_m^i(s(t))$  is the output of the  $i$ th sample of the strategy model at time  $t$  with system state  $s(t)$ , and  $a_h^i(t)$  is the output of the  $i$ th sample of the historical data of human decision-making at time  $t$ .

Finally, we give the optimization algorithm of human-machine traded control, including MTHA-B and HTMA-B. In this article, on the basis of solving the problem of ordinary human-machine traded control for SDM (1), it incorporates the concept of autonomous boundary (which may be human boundary or machine boundary). The optimization goal of the algorithm is not only to optimize the strategy function directly related to the decision-making behavior, but also to learn to optimize the autonomous boundary that indirectly affects the decision-making behavior. First, we initialize the corresponding autonomous boundary information, policy network and its parameters. During the dynamic evolution of the system, human partners and intelligent machines will, respectively, give

$$a(t) = f^a(a_h(t), a_m(t), b_h(t))$$

$$= \begin{cases} \text{Machine: } \mathbf{a}_m(\mathbf{t}), \{c_m(t) \geq c_h(t)\} \& \{J_{h,m}(s(t), a_m(t)) \geq \max\{J_{h,m}(s(t), a_h(t)), J_{h,m}(s(t), b_h(t-1))\}\} \\ \text{Boundary: } \mathbf{b}_h(\mathbf{t}-1), \{c_h(t) \geq c_m(t)\} \& \{J_{h,m}(s(t), b_h(t-1)) \geq \max\{J_{h,m}(s(t), a_h(t)), J_{h,m}(s(t), a_m(t))\}\} \\ \text{Human: } \mathbf{a}_h(\mathbf{t}), & \text{otherwise} \end{cases} \quad (10)$$

$$a(t) = f^a(a_h(t), a_m(t), b_m(t))$$

$$= \begin{cases} \text{Human: } \mathbf{a}_h(\mathbf{t}), \{c_h(t) \geq c_m(t)\} \& \{J_{h,m}(s(t), a_h(t)) \geq \max\{J_{h,m}(s(t), a_m(t)), J_{h,m}(s(t), b_m(t-1))\}\} \\ \text{Boundary: } \mathbf{b}_m(\mathbf{t}-1), \{c_m(t) \geq c_h(t)\} \& \{J_{h,m}(s(t), b_m(t-1)) \geq \max\{J_{h,m}(s(t), a_h(t)), J_{h,m}(s(t), a_m(t))\}\} \\ \text{Machine: } \mathbf{a}_m(\mathbf{t}), & \text{otherwise} \end{cases} \quad (11)$$



---

**Algorithm 3:** Machine Traded Human Control Based on Autonomous Boundary (MTHA-B).
 

---

- 1: **Initialization:** Randomly initialize the intelligent machine decision network  $p_m$  and its parameters; Initialize the autonomous boundary of human  $\bar{B}_h$ .
  - 2: **Input:** The system state  $s(t)$ .
  - 3: **Output:** The final decision behavior  $a(t)$ .
  - 4: **repeat**
  - 5:     The intelligent machine decision network calculate machine behavior  $a_m(t)$  according to  $p_m(\cdot)$ , and its credibility  $c_m(t)$  based on Monte Carlo estimation (12).
  - 6:     Input human behavior  $a_h(t)$  at time  $t$  through peripherals (such as mouse, keyboard, joystick, etc.), and calculate its credibility  $c_h(t)$  based on Monte Carlo estimation (13).
  - 7:     Obtain the final decision behavior  $a(t)$  at time  $t$  based on the arbitration formula (10).
  - 8:     According to algorithm 1, update the autonomous boundary information  $b_h(t)$  at time  $t$ .
  - 9: **until** Reaching maximum training time step  $N$ .
- 

decision-making behaviors  $a_h(t)$  and  $a_m(t)$  for real-time state  $s(t)$ , and the corresponding decision-making credibility evaluation  $c_h(t)$  and  $c_m(t)$ . Considering the common objective function of machine traded control, based on the learning method of the autonomous boundary in (4) and algorithm 1, the final decision-making behavior  $a(t)$  is calculated according to arbitration (10). The HTMA-B algorithm can be obtained similarly. The difference from MTHA-B is that  $\bar{B}_h$  in step 1 of the algorithm 3 needs to be replaced with  $\bar{B}_m$ , and the arbitration function in step 7 needs to be replaced with (11). Finally, the autonomous boundary update at the current time  $t$  is completed. Repeat until the end of the training.

#### IV. EXPERIMENTS

In this section, two experiments will be conducted to verify the optimization methods proposed in this article, namely, machine traded control, human traded control.

##### A. Machine Traded Control

Aiming at the problem of machine traded control, this subsection conducts simulation experiments on the basis of reinforcement learning. Specifically, we use the LunarLander in OpenAI Gym, as shown in Fig. 3. During the descent of the lander, if the lander crashes or comes to a standstill, a complete experience ends, and a reward of  $-100$  or  $100$  is awarded. Each leg of the lander touches the ground with a reward of  $10$ . When the main engine is turned ON, it consumes fuel with a reward of  $-0.3$  per frame (assuming that the fuel is unlimited). The state vector of lander  $s(t)$  includes: coordinates  $(x(t), y(t))$ , speed  $(\dot{x}(t), \dot{y}(t))$ , angle  $(\theta(t), \dot{\theta}(t))$ , whether to land ( $leg_l(t), leg_r(t)$ ), and landing point coordinates  $h(t)$ .

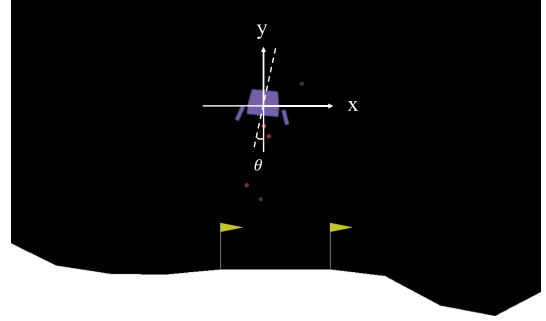


Fig. 3. Simulation environment of the LunarLander.

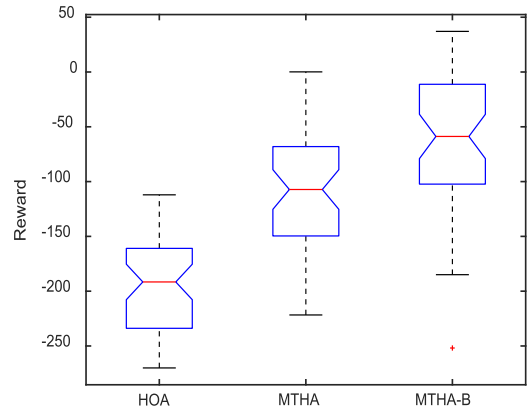


Fig. 4. Mean reward comparison of algorithms HOA, MTHA, and MTHA-B for 500 episodes.

This experiment uses the DQN algorithm as the machine agent algorithm. We conduct comparative experiments on human individual control and machine traded control, including rewards comparison, success rates comparison, crash rates comparison, percentage of human-machine actions, action correspondence of human-machine, and trajectories comparison. And before the start of the formal comparison experiment, we first pretrain the machine agent algorithm DQN so that the machine agent involved in human control has a certain reasonable decision-making ability. In the figures shown below, we use human-only-algorithm (HOA) to indicate that only human operators will control. In this article, MTHA refers to the intervention of DQN machine agent strategy on human operations, which corresponds to the MTHA in Section III-B. MTHA-B represents the optimization algorithm that adds boundary information on the basis of MTHA, which corresponds to the MTHA-B in Section III-B. In more detail, we define and judge the autonomous boundary of human, and apply this boundary information to the control optimization algorithm of machine intervene human, so as to achieve the goal of improving decision-making performance.

As shown in Figs. 4 and 5(a), the algorithm HOA has the worst reward, which is in line with our conjecture that the accuracy of human control is not high. In contrast, machine traded control algorithms (MTHA and MTHA-B) can increase cumulative rewards to varying degrees. And in the machine traded control algorithm, MTHA-B has a better effect due to the use of additional boundary information. The accumulative



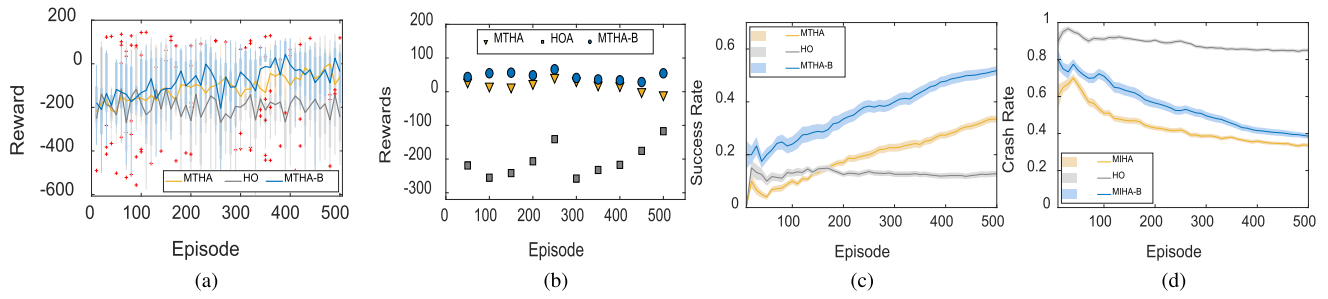


Fig. 5. Comparison of algorithm HOA, MTHA, and MTHA-B. (a) Rewards: The solid line in the figure indicates the mean value of the reward, the red plus sign indicates abnormal points, and the shadow indicates the box area where most points fall. (b) Rewards of success episode. (c) Success rates: the solid line in the figure represents the mean value of the success rates, and the shade represents uncertainty. (d) Crash rates: The solid line in the figure represents the mean value of the crash rates, and the shade represents uncertainty.

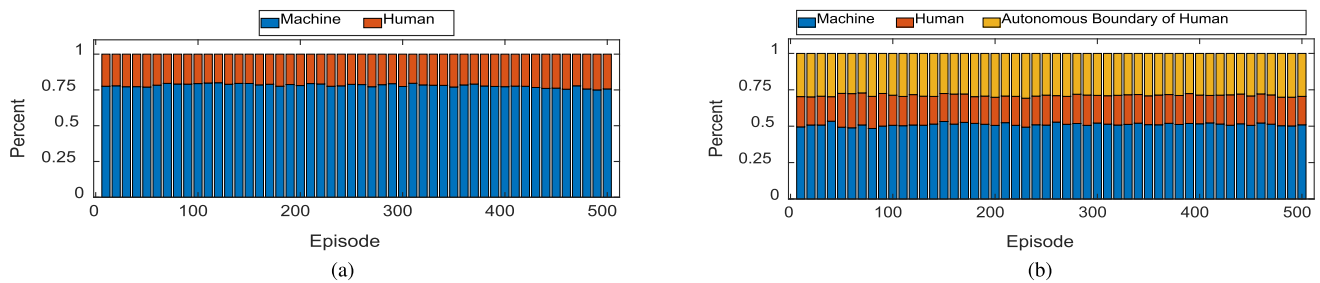


Fig. 6. Percentage of human actions  $a_h$  and machine actions  $a_m$ . (a) MTHA. (b) MTHA-B.

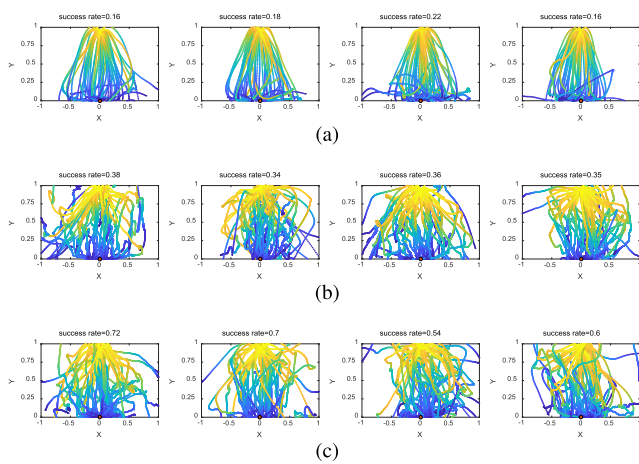


Fig. 7. Trajectories comparison of algorithm. (a) HOA. (b) MTHA. (c) MTHA-B.

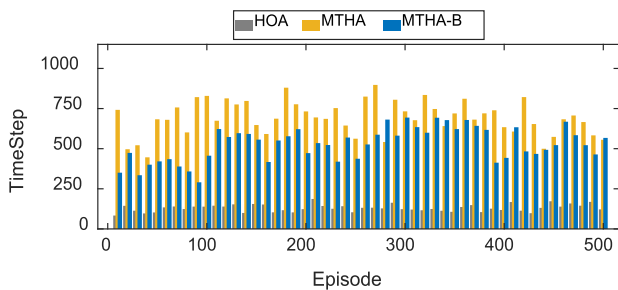


Fig. 8. Timesteps in each episode of the algorithm HOA, MTHA, and MTHA-B.

rewards here refer to the accumulative reward value obtained by each complete episode. In order to avoid the contingency of the experimental effects, we randomly collected 500 experimental results to evaluate the mean and uncertainty. In order to prove the superiority of the algorithm MTHA-B, we averaged the experience rewards of successful landing in 500 episodes, and obtained Fig. 5(b). Combining Fig. 5(a) and (b), we observe that the algorithm MTHA-B not only has an advantage in the rewards of the overall episodes but also makes the successful landing episodes have higher rewards than HOA and MTHA. Therefore, the experimental results in this section are convincing and effective to prove the optimal design of machine traded control.

In the LunarLander, landing on the landing site smoothly and safely is the decisive factor for the success of the game. Next, we compare the landing success rates and crash rates of the algorithm HOA, MTHA, and MTHA-B. For the success rates in Fig. 5(c),  $MIHA-B > MIHA > HOA$ . The success rates of the algorithm HOA are continuously low, which stems from humans' low-precision operations for not good at work, as well as the weakness of the required learning time and response ability. With the trade of the machine, the success rates have been significantly improved. In particular, in the landing success rates, with the increase of episodes, the algorithm MTHA-B can continuously increase the success rates to 0.55 or even higher. Similarly, for the crash rates shown by Fig. 5(d), the sharply reduced crash rates are due to the machine agents that make decision-making more precise and robust. However, we found that the crash rates of MTHA-B are higher than that of

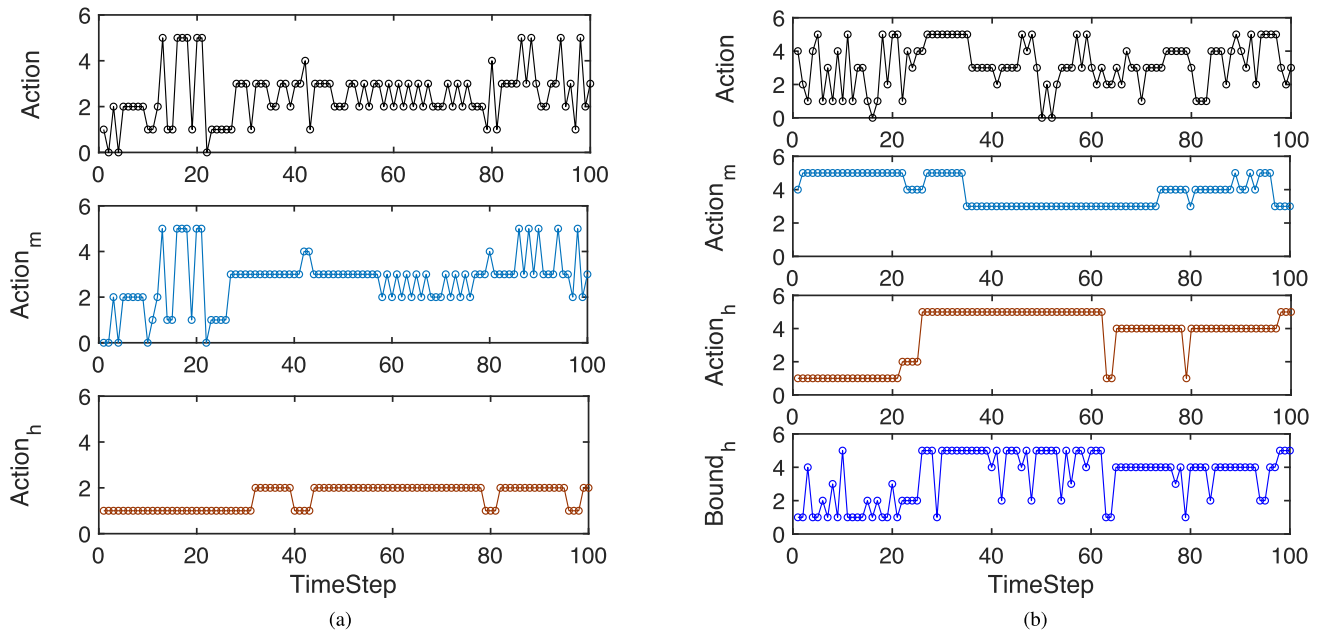


Fig. 9. Actions correspondence of algorithm. (a) MTHA: the action correspondence of the first 100 time steps; from top to bottom, there are final actions, machine actions, and human actions. (b) MTHA-B: the action correspondence of the first 100 time steps; from top to bottom, there are final actions, machine actions, human actions, and boundaries of human.

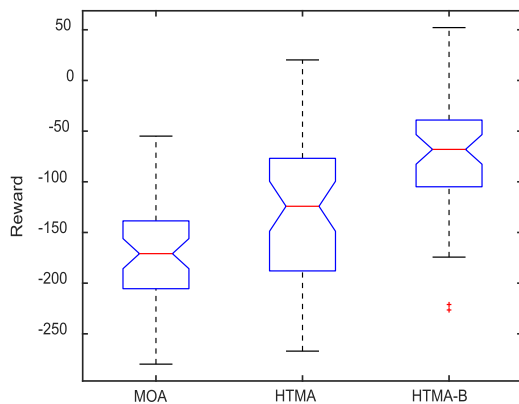


Fig. 10. Mean reward comparison of algorithms MOA, HTMA, and HTMA-B for 500 episodes.

MTHA, which seems to be a bad-looking signal. In fact, this is also related to our use of autonomous boundary information. It is a clear to measure the timing of trade through boundary information, but some of the instability caused by boundary introduction cannot be ignored. This requires developers to make a compromise between success rates and moderate instability.

From above Fig. 5, we observe that with the continuous improvement of machine decision-making capabilities, humans can gradually hand over tasks that they are not good at to intelligent machines to complete, or part of them to complete, as in the topic of this section, i.e., the machine intervenes in human control. This can also be derived from the behavior percentages in Fig. 6. It can be seen from the figure that the proportion of human actions in MTHA is relatively low. In MTHA-B, the final decision-making action is a combination of

human actions, machine actions, and human boundaries. More specifically, Fig. 6(b) is the percentage of times each of the three components actions (the red human behavior, the yellow human autonomous boundaries, and the blue machine decision-making behavior) were chosen by the arbitrator (10) in episodes. From Fig. 6(b), it can be found that human actions, human boundaries, and machine actions affect the final decision-making behavior in a ratio of 2 : 3 : 5. This is understandable and in line with our definition of the boundary because it is possible to achieve the best on the boundary. In this section, we mainly consider the upper bound of human autonomy, which means that no machine intervention is required when the credibility of human decision-making behavior is high. At this time, we only need to make a selection on the boundary between human real-time decision input and autonomous boundary of human.

Next, we compare the landing trajectories of HOA, MTHA, and MTHA-B algorithms, as shown in Fig. 7. First of all, the landing trajectory of HOA looks clean and neat, but combined with its low success rates and high crash rates (as shown in Fig. 11) and timesteps (see Fig. 8), we can conclude that HOA tends to crash directly and rapid failure caused by the imprecision of human operations. Second, we found the landing trajectories of MTHA algorithm more messy but the success rates have been improved, which is due to the machine traded control. The messiness of trajectories of MTHA-B algorithm is between that of HOA and MTHA, and the success rates and running time steps of MTHA-B are greatly improved, which is more in line with the algorithm goal (to complete the task better and faster). About better and faster, it is also reflected in the running time step of the algorithms. The algorithm MTHA has the longest running time step, which is related to the larger proportion of machine actions [see Fig. 6(a)]. That is, the machine trades a

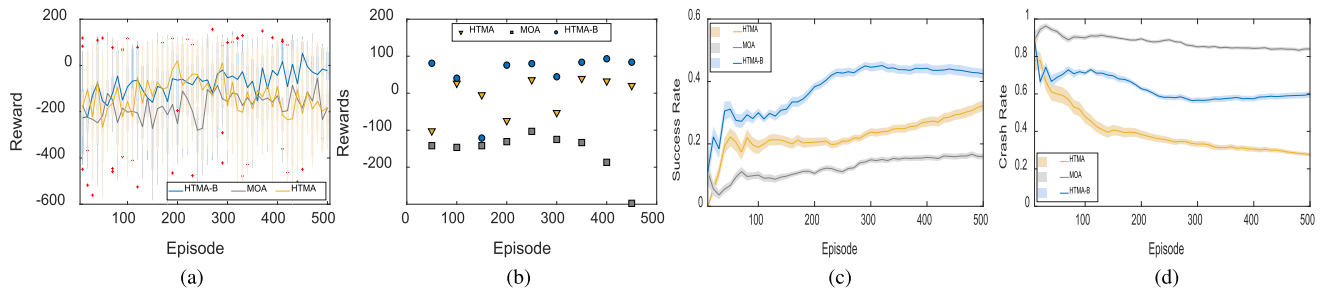


Fig. 11. Comparison of algorithm MOA, HTMA, and HTMA-B. (a) Rewards: The solid line in the figure indicates the mean value of the reward, the red plus sign indicates abnormal points, and the shadow indicates the box area where most points fall. (b) Rewards of success episodes. (c) Success rates: the solid line in the figure represents the mean value of the success rates, and the shade represents uncertainty. (d) Crash rates: The solid line in the figure represents the mean value of the crash rates, and the shade represents uncertainty.

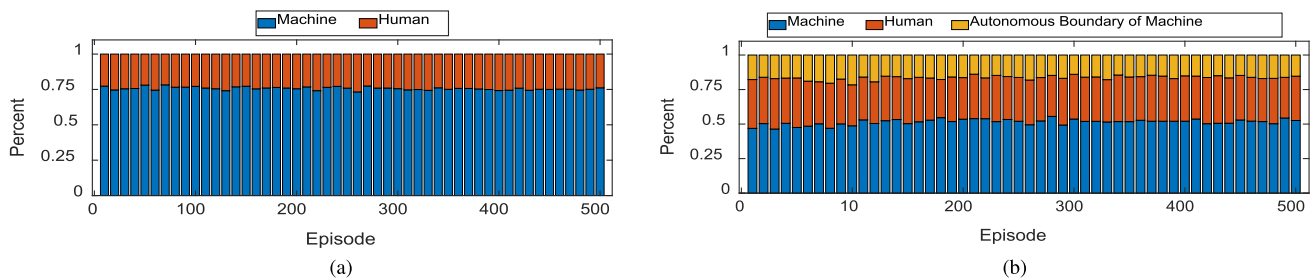


Fig. 12. Percentage of human actions  $a_h$  and machine actions  $a_m$ . (a) HTMA. (b) HTMA-B.

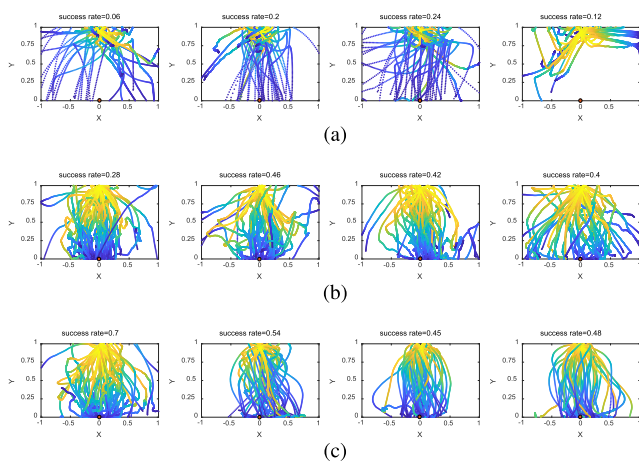


Fig. 13. Trajectories comparison of algorithms: (a) MOA; (b) HTMA; (c) HTMA-B.

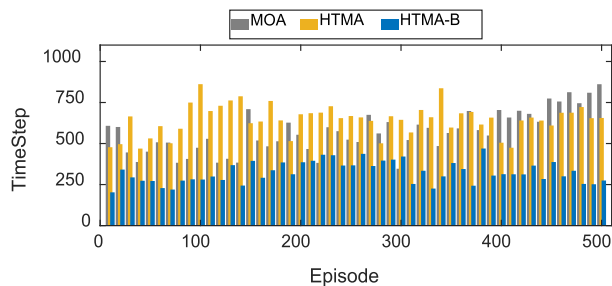


Fig. 14. Timesteps in each episode of the algorithms MOA, HTMA, and HTMA-B.

slower learning rate for a partial increase in the success rates. The running time of the algorithm MTHA-B is between the three algorithm. In other words, MTHA-B not only improves the success rates, reduces the crash rates, but also speeds up the task completion speed.

Finally, in order to facilitate the understanding of the execution process of the machine traded control in the human control systems mentioned in this subsection, we give the final decision-making behavior  $a(t)$ , the machine decision-making behavior  $a_m(t)$ , and the corresponding relationship between the human decision-making behavior  $a_h(t)$ , as shown in Fig. 9. From Fig. 9(a), we observe that the value of the final decision-making behavior  $a(t)$  is always between the machine decision-making behavior  $a_m(t)$  and the human decision-making behavior  $a_h(t)$ . From Fig. 9(b), the final action  $a(t)$  is to choose between  $a_m(t)$ ,  $a_h(t)$ , and  $b_h(t-1)$ , where  $b_h(t-1)$  is the boundary information of human, and we add additional decision boundary information to optimize and judge the final decision signal.

## B. Human Traded Control

In this section, LunarLander is still used, and DQN is used as the machine agent algorithm. But we put more emphasis on human traded control in this section. We conduct comparative experiments between the machines' independent control and human traded control, including reward comparison, success rates comparison, crash rates comparison, percentages of human-machine actions, actions correspondence of human-machine, and landing trajectories comparison. Similarly, before the formal

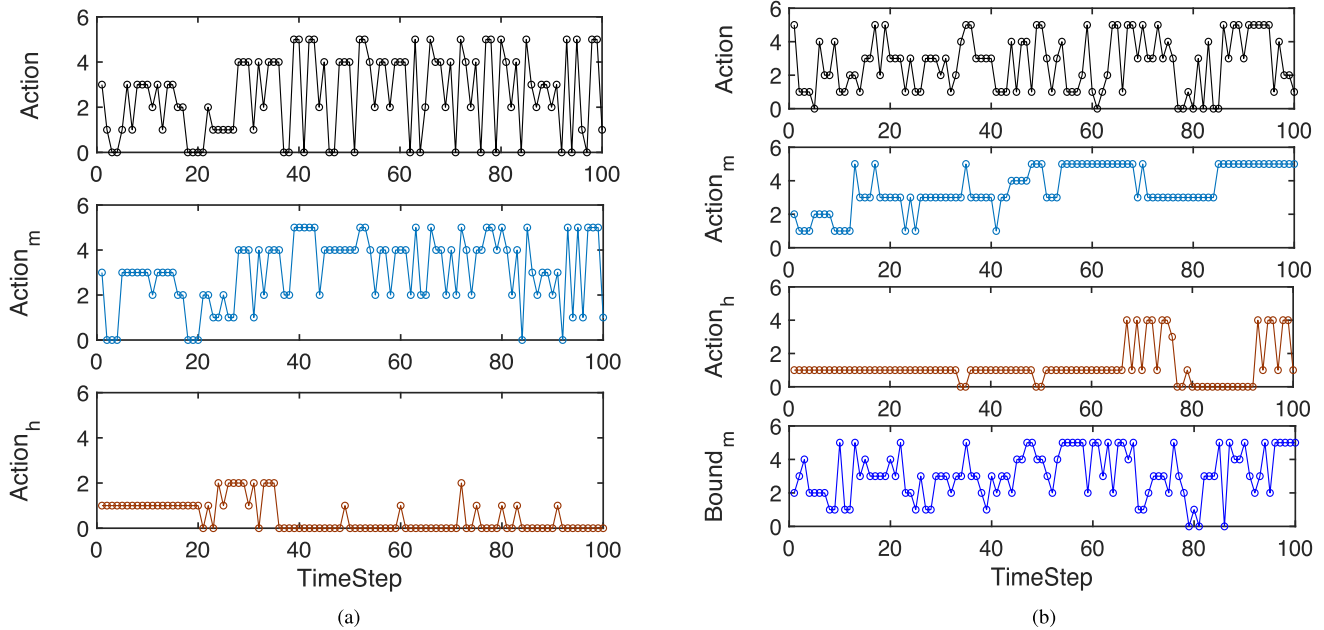


Fig. 15. Actions correspondence of algorithm. (a) HTMA: the action correspondence of the first 100 time steps; from top to bottom, there are final actions, machine actions, and human actions. (b) HTMA-B: the action correspondence of the first 100 time steps; from top to bottom, there are final actions, machine actions, human actions, and boundaries of machine.

comparative experiment, we first pretrain the machine agent algorithm DQN. In the following figures in this section, the use of machine-only-algorithm (MOA) indicates that only machine control is used. MTHA refers to the intervention of human operations on DQN machine agent strategy, which corresponds to the HTMA in Section III-B. HTMA-B represents the optimization algorithm that adds boundary information on the basis of HTMA, which corresponds to the HTMA-B in Section III-B. In more detail, we define and judge the autonomous boundary of machine, and apply this boundary information to the control optimization algorithm of human intervene machine, so as to achieve the goal of improving decision-making performance.

In Figs. 10 and 11(a), comparing MOA, HTMA, and HTMA-B, within a certain period of time, the human intervention algorithm (HTMA, HTMA-B) is more rewarding than the machine-only algorithm MOA. Particularly, the algorithm HTMA-B has a significant improvement in rewards, which also confirms the improvement of the decision-making performance of the autonomous boundary we described. In addition, to be more specific, we select the successful landing episodes from 500 random episodes, and average their rewards to get Fig. 11(b). In Fig. 11(b), we observe that the rewards corresponding to successful landing episodes confirm to the relationship of HTMA-B > HTMA > MOA. Therefore, we can conclude that the method proposed in this article (HTMA-B) has greater advantages compared to the previous machine autonomy (MOA) and human trade machine control (HTMA).

In the LunarLander, landing on the landing site smoothly and safely is the decisive factor for the success of the game. Next, we compare the landing success rates and crash rates of the algorithm HOA, MTHA, and MTHA-B. For the success rates in Fig. 11(c), the success rates of MOA continues to

below, due to the fact that the machine agent is based on neural network training, which takes a lot of time to train to complete. With human traded control, the landing success rates have been improved, especially the method described in this article HTMA-B can continue to increase the landing success rates to 0.45 or even higher. Regarding the crash rates in Fig. 11(d), our experimental result is MOA > HTMA-B > HTMA. Similar to HOA > MTHA-B > MTHA in the previous section, we attribute this phenomenon to the use of autonomous boundary information. But it is undeniable that the optimal design based on autonomous boundary information does have a significant improvement effect on decision-making performance (such as reward value, success rates), so researchers need to make a better tradeoff or compromise. From the above experimental results on rewards, landing success rates, and crash failure rates, we can see that machine agents can use some of their autonomy to reduce human labor. However, the current research stage of the neural network-based machine agent still needs improvement in learning ability, so it is necessary to seek a better human traded control strategy, such as the human traded control design based on autonomous boundary (HTMA-B) in this section, which is meaningful and effective.

Fig. 12 describes the action percentage of the algorithms HTMA, HTMA-B. We observe that in the scenario of human trade in the machine, human actions have a higher share, which is directly related to the priority level. Human actions, machine boundaries, and machine actions in HTMA-B affect the final decision-making behavior in an ratio of 3 : 1 : 4, respectively. Compared with 2 : 3 : 5 in MTHA-B [see Fig. 6(b)], the proportion of human actions has increased. This is due to two reasons: the attributes of the human traded algorithm are determined; the efficiency of human traded control to train better machine agents



is improved. In addition, we find that machine actions accounted for about 50% in both MTHA-B and HTMA-B, which also aroused our thinking. That is, whether it is human traded control or machine traded control, in the real-time dynamic evolution process, the machine is a decision-making subject that cannot be ignored, which is determined by the inherent attributes of the example itself that are more suitable for machine manipulation.

Next, we compare the landing trajectories of the algorithms MOA, HTMA, and HTMA-B, as shown in Fig. 13. We find that in Fig. 13(a) corresponding to the MOA, the landing trajectory is not as orderly as in the algorithm HOA diagram 7(a) (indicating the learning effect of the machine agent), but the success rates is very low. In Fig. 13(b), the algorithm HTMA has made preliminary improvements to MOA, including the success rates and the degree of divergence. Furthermore, in Fig. 13(c), the algorithm HTMA-B enhances HTMA, which not only improves the success rates, but also has a more orderly and rapid landing trajectory. Compared with the machine traded control in Fig. 8, in the human traded control in Fig. 14, the running time steps of the algorithms MOA and HTMA are not much different. However, the running time steps of the algorithm HTMA-B have been significantly reduced, which corresponds to the increase in the success rates and the decrease in the crash rates in Fig. 11. Therefore, HTMA-B has the effect of significantly improving decision-making performance in terms of running time step.

Finally, in order to facilitate the understanding of the execution process of the human trade in the machine control systems mentioned in this section, we give the final decision-making behavior  $a(t)$ , the machine decision-making behavior  $a_m(t)$ , and the corresponding relationship between the human decision-making behavior  $a_h(t)$ , as shown in Fig. 15. From Fig. 15(a), we observe that the value of the final decision-making behavior  $a(t)$  is always between the machine decision-making behavior  $a_m(t)$  and the human decision-making behavior  $a_h(t)$ . From Fig. 15(b), the final action  $a(t)$  is to choose between  $a_m(t)$ ,  $a_h(t)$ , and  $b_m(t-1)$ , where  $b_m(t-1)$  is the boundary information of machine, and we add additional decision boundary information to optimize and judge the final decision signal.

## V. CONCLUSION

This article considered a kind of traded control of human on the loop to solve the problem of human-machine SDM. Different from the previous traded control, this article used the autonomous boundary to optimize the design of the decision-making systems, as shown in Fig. 2. We first discussed the autonomous boundary decision method in the human-machine traded control systems, and then optimized the design decision system based on this boundary information. There are two dynamic optimization goals in the decision-making process: the machine strategy network directly related to the decision-making behavior; the autonomous boundary information indirectly related to the decision-making behavior. The arbitration mechanism designed in this article not only evaluated the credibility of decision-making behavior but also added additional autonomous boundary information. These have a certain degree of effect whether they are out of control of machines or humans. Therefore, the decision-making performance finally presented

could be improved and enhanced. We conducted simulation experiments on machine traded control and human traded control, respectively, and the experimental results verified the effectiveness of the methods described in this article (MTHA-B and HTMA-B).

In future work, we will continue to apply autonomous boundary information to the field of shared control, so that the human-machine hybrid intelligent control systems have a more complete solution, and the humans and machines have a relatively clear decision boundary. In addition, we will consider more complex situations, such as the case where the machine agent does not understand the task goal. Then, we need to make intentional reasoning by observing human behavior, and combine intentional reasoning information with autonomous boundary information to get a better final decision, etc.

## REFERENCES

- [1] P. David *et al.*, "Computational intelligence, a logical approach," *Mañá Alvarado*, 1999.
- [2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 2003.
- [3] H. S. Chang, M. C. Fu, J. Hu, and S. I. Marcus, "Google deep mind's alphago," *OR/MS Today*, vol. 43, no. 5, pp. 24–29, Oct. 2016.
- [4] P. Hamet and J. Tremblay, "Artificial intelligence in medicine," *Metabolism*, vol. 69, pp. S36–S40, Apr. 2017.
- [5] J.-C. Latombe, *Robot Motion Planning*. Norwell, MA, USA: Kluwer, 1991.
- [6] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *Int. J. Robot. Res.*, vol. 32, no. 7, pp. 790–805, Jun. 2013.
- [7] D. Sadigh, S. S. Sastry, and S. A. Seshia, "Verifying robustness of human-aware autonomous cars," *IFAC-PapersOnLine*, vol. 51, no. 34, pp. 131–138, Dec. 2018.
- [8] C. P. Lam, "Improving sequential decision making in human-in-the-loop systems," Ph.D. dissertation, Dept. Eng. Elect., Univ. California, Berkeley, Berkeley, CA, USA, 2017.
- [9] K. Fuchs, "Minimally invasive surgery," *Endoscopy*, vol. 34, no. 2, pp. 154–159, 2002.
- [10] O. Alagoz, H. Hsu, A. J. Schaefer, and M. S. Roberts, "Markov decision processes: A tool for sequential decision making under uncertainty," *Med. Decis. Making*, vol. 30, no. 4, pp. 474–483, Jul. 2010.
- [11] J. Verdura, M. E. Carroll, R. Beane, S. Ek, and M. P. Callery, "Systems methods and instruments for minimally invasive surgery," US Patent 6,165,184, Dec. 26, 2000.
- [12] R. Michelmore, M. Kwiatkowska, and Y. Gal, "Evaluating uncertainty quantification in end-to-end autonomous driving control," 2018, *arXiv:1811.06817*.
- [13] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, New York City, NY, USA, 2016, pp. 1050–1059.
- [14] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.
- [15] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," presented at the Int. Conf. Learn. Representations, May 2016. [Online]. Available: <https://arxiv.org/abs/1509.02971>
- [16] V. Mnih *et al.*, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, New York City, NY, USA, 2016, pp. 1928–1937.
- [17] M. Hessel *et al.*, "Rainbow: Combining improvements in deep reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 3215–3222.
- [18] Y. Gal, R. McAllister, and C. E. Rasmussen, "Improving pilco with Bayesian neural network dynamics models," in *Proc. Int. Conf. Mach. Learn.*, New York City, NY, USA, 2016, Paper 25.
- [19] J. C. G. Higuera, D. Meger, and G. Dudek, "Synthesizing neural network controllers with probabilistic model-based reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 2538–2544.
- [20] A. Kendall *et al.*, "Learning to drive in a day," in *Proc. IEEE Int. Conf. Robot. Automat.*, Montreal, QC, Canada, 2019, pp. 8248–8254.

- [21] P. Owan, J. Garbini, and S. Devasia, "Addressing agent disagreement in mixed-initiative traded control for confined-space manufacturing," in *Proc. AIM*, 2017, pp. 227–234.
- [22] C. Phillips-Grafflin *et al.*, "From autonomy to cooperative traded control of humanoid manipulation tasks with unreliable communication," *J. Intell. Robot. Syst.*, vol. 82, no. 3, pp. 341–361, Jun. 2016.
- [23] B. M. Muir, "Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems," *Ergonomics*, vol. 37, no. 11, pp. 1905–1922, Nov. 1994.
- [24] R. Bellman, "A markovian decision process," *J. Math. Mechanics*, vol. 6, no. 5, pp. 679–684, 1957.
- [25] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Hoboken, NJ, USA: Wiley, 2014.
- [26] L. Fridman, L. Ding, B. Jenik, and B. Reimer, "Arguing machines: Human supervision of black box AI systems that make life-critical decisions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1335–1343.
- [27] A. Broad, T. Murphey, and B. Argall, "Highly parallelized data-driven MPC for minimal intervention shared control," presented at the Conf. Robot. Sci. Syst., Jun. 2019. [Online]. Available: <https://arxiv.org/abs/1906.02318>
- [28] Y. Oh, M. Toussaint, and J. Mainprice, "Learning arbitration for shared autonomy by hindsight data aggregation," 2019, *arXiv:1906.12280*.
- [29] A. A. Allaban, V. Dimitrov, and T. Padir, "A blended human-robot shared control framework to handle drift and latency," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot.*, 2019, pp. 81–87.
- [30] Z. Lin, B. Harrison, A. Keech, and M. O. Riedl, "Explore, exploit or listen: Combining human feedback and policy model to speed up deep reinforcement learning in 3D worlds," 2017, *arXiv:1709.03969*.
- [31] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," presented at the Conf. Robot.-Sci. Syst., Jun. 2018. [Online]. Available: <https://arxiv.org/abs/1802.01744>
- [32] J. Tjomsland, A. Shafti, and A. A. Faisal, "Human-robot collaboration via deep reinforcement learning of real-world interactions," 2019, *arXiv:1912.01715*.
- [33] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. New York, NY, USA: Springer, 2012.
- [34] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. Int. Conf. Mach. Learn.*, Lille, France, 2015, pp. 1613–1622.
- [35] S. Depeweg, J. M. Hernández-Lobato, F. Doshi-Velez, and S. Udluft, "Learning and policy search in stochastic dynamical systems with bayesian neural networks," presented at the Int. Conf. Learn. Representations, Apr. 2016. [Online]. Available: <https://arxiv.org/abs/1605.07127>
- [36] Y. Gal, "Uncertainty in deep learning," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2016.
- [37] B. Lötjens, M. Everett, and J. P. How, "Safe reinforcement learning with model uncertainty estimates," in *Proc. IEEE Int. Conf. Robot. Automat.*, Montreal, QC, Canada, 2019, pp. 8662–8668.