Human-Centered Reinforcement Learning: A Survey

Guangliang Li, *Member, IEEE*, Randy Gomez, Keisuke Nakamura, *Member, IEEE*, and Bo He, *Member, IEEE*

***Abstract*—Human-centered reinforcement learning (RL), in which an agent learns how to perform a task from evaluative feedback delivered by a human observer, has become more and more popular in recent years. The advantage of being able to learn from human feedback for a RL agent has led to increasing applicability to real-life problems. This paper describes the state-of-the-art human-centered RL algorithms and aims to become a starting point for researchers who are initiating their endeavors in human-centered RL. Moreover, the objective of this paper is to present a comprehensive survey of the recent breakthroughs in this field and provide references to the most interesting and successful works. After starting with an introduction of the concepts of RL from environmental reward, this paper discusses the origins of human-centered RL and its difference from traditional RL. Then we describe different interpretations of human evaluative feedback, which have produced many human-centered RL algorithms in the past decade. In addition, we describe research on agents learning from both human evaluative feedback and environmental rewards as well as on improving the efficiency of human-centered RL. Finally, we conclude with an overview of application areas and a discussion of future work and open questions.**

***Index Terms*—Human agent/robot interaction, human reward, interactive reinforcement learning (RL), interactive shaping, policy shaping.**

I. Introduction

REINFORCEMENT learning (RL) [1] is a framework in which an agent or robot system can learn how to perform a sequential decision task [2] online by interacting with the environment. Specifically, in RL, at the beginning of the learning process, an agent observes an environmental state, based on which it selects an action from a finite set of actions and performs it in the task. The agent then receives an environmental reward signal from a reward function predefined by the agent designer. Such reward indicates the quality of the agent's action. Then the agent will transition to a new environmental state based on the current state-action pair. In the new state, the agent will select and execute another action, and receive another new reward signal. The cycle of observing environmental state, selecting action, receiving reward, and transitioning to a new state is repeated until the learning is finished. The objective is to learn an optimal policy that receives the most cumulative reward. Such a policy decides and selects a sequence of actions for states encountered by the agent in the environment.

RL has proven to be an effective robot learning method and been successfully applied in many task domains [3]. However, the learning speed of a standard RL agent is very slow and it usually takes a very long time for the agent to explore with trial and error before achieving the task goal for the first time. For real-world robot learning, the failure at the beginning of the learning stage may have a large cost. Therefore, reward shaping [4] was developed to improve the agent's learning performance in complex tasks. Reward shaping works by adding a supplemental reward via a potential function to the environmental reward. The potential function is usually predefined by the agent designer and can also be learned [5]–[7].

However, with increasing attention being paid to personalized and service robots, autonomous agents will have the potential to be applied in the real world and operate in human living environments in the near future. Therefore, agents and humans will be closely connected to each other, and the interaction between them will increase to a great extent. In real-world applications, agents not only need to learn how to perform tasks, humans may also want to change the agents' optimal behavior by teaching them interactively according to their likings. In this case, standard RL cannot be applied in real-world agents that learn from human beings even with reward shaping, since the optimal behavior is usually preprogrammed via a reward function and most human users are laymen in agent design and programming.

Inspired by work on reward shaping, human-centered RL has been developed and proven to be a powerful method for facilitating ordinary people to teach agents in a natural way. A human user might not be an expert in programming but has much knowledge about how to perform a task, which will reduce the agent's exploration time and speed up its learning. Similar to standard RL, the central problem of human-centered RL is addressing the challenge of human-delivered evaluative feedback. Most human-centered RL algorithms are distinguished by interpretations of human feedback. Ho *et al.* [8] examine how teachers provide evaluative feedback in response to actions of a learner, and compare the reward-maximizing model based on standard RL and action-feedback model based on research on communicative intent.

Manuscript received June 16, 2017; revised March 7, 2018 and November 23, 2018; accepted March 20, 2019. Date of publication May 7, 2019; date of current version July 13, 2019. This work was supported in part by the Natural Science Foundation of China under Grant 51809246, in part by the Natural Science Foundation of Shandong Province under Grant ZR2018QF003, in part by the Fundamental Research Funds for the Central Universities under Grant 841713015, in part by the China Postdoctoral Science Foundation under Grant 861705020036, and in part by the Honda Research Institute Japan Co. Ltd. This paper was recommended by Associate Editor F. Ritter. *(Corresponding author: Guangliang Li.)*

G. Li and B. He are with the College of Information Science and Engineering, Ocean University of China, Qingdao 266100, China (e-mail: guangliangli@ouc.edu.cn; bhe@ouc.edu.cn).

R. Gomez and K. Nakamura are with the Honda Research Institute Japan Co., Ltd., Wako 351-0114, Japan (e-mail: r.gomez@jp.honda-ri.com; k.nakamura@jp.honda-ri.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/THMS.2019.2912447

Because of the growing popularity and recent developments in the field of human-centered RL, it deserves a survey on its own right though several surveys on RL already exist [9]–[12]. The objective of this survey is to give an overview of work on human-centered RL and technical details of some representative algorithms, and also to discuss some open problems to be solved in this area. Depending on the interpretation of human evaluative feedback, there are mainly three kinds of human-centered RL algorithms developed that facilitate agents to learn from them: interactive shaping (also called learning from human reward), learning from categorical feedback, and learning from policy feedback. The focus of this paper is put on these human-centered RL algorithms that deal with different interpretations of human evaluative feedback and work on learning from both human feedback and environmental reward. It also provides an overview of work on improving an agent's learning from human evaluative feedback. Additionally, some application papers are presented and discussed.

The rest of this paper starts with an introduction of basic concepts in RL in Section II, which is the cornerstone of human-centered RL. Section III surveys human-centered RL algorithms and methods for learning from both human evaluative feedback and environmental reward. Section IV describes studies and methods for improving the efficiency of learning from human feedback. Section V describes applications of human-centered RL. A discussion and outlook is provided in Section VI. Finally, Section VII concludes.

II. Reinforcement Learning

In RL, an agent learns how to perform a sequential decision task, i.e., a policy that decides which action to take in a state of the environment the agent encounters. A sequential decision task is modeled as a Markov Decision Process (MDP), denoted as $\{S, A, T, R, \gamma\}$. In MDP, S represents a set of all possible states and A represents a set of all possible actions. Time is divided into discrete time steps. At each time step t, the agent receives a representation of the environmental state, $s_t \in S$, takes an action $a_t \in A$ that results in next state of the environment s_{t+1}. One time step later, as a consequence of the action a_t taken based on the current state s_t, the agent will receive a numerical reward, r_{t+1} specified by the reward function $R : S \times A \times S \rightarrow \Re$, which decides a numeric reward value at each time step based on the current state, action chosen, and the resultant next state. The probability of next state s_{t+1} that the agent will experience is decided by the transition function $T : S \times A \times S$, which describes the probability of transitioning from one state to another given a specific action, $T(s_t, a_t, s_{t+1}) = Pr(s_{t+1}|s_t, a_t)$. Fig. 1 depicts an agent's learning by interacting with the environment in a standard RL framework.

The agent's learned behavior is described as a policy, $\pi : S \times A$, where $\pi(s, a) = Pr(a_t = a|s_t = s)$ is the probability of selecting a possible action $a \in A$ in a state s. The goal of the agent is to maximize the accumulated reward the agent receives, denoted as $\sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$ at time step t, where γ is the discount factor (usually $0 \leq \gamma < 1$). γ determines the present value of rewards received in the future: a reward received k time steps in the future is worth only γ^{k-1} times what it would be worth if it were received immediately. The return for a policy π is denoted as $\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, \pi(s_{t+k}), s_{t+k+1})$. There are usually two associated value functions for each learned policy π. One is the $state\text{-}value\ function$, referred to as the value of a state, $V^\pi(s)$, which is the expected return when an agent starts in a state s and follows a policy π thereafter, where

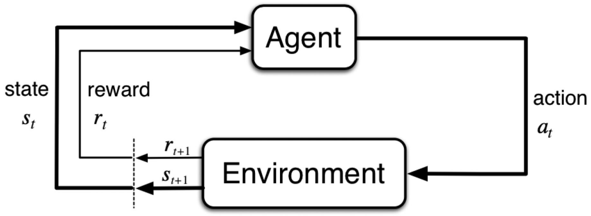


Fig. 1. Agent learning by interacting with the environment in the standard reinforcement learning framework (reproduced from [1]).

$$V^\pi(s) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right]. \quad (1)$$

Similarly, another value function is the $action\text{-}value\ function$, referred to as the value of a state-action pair, $Q^\pi(s, a)$, which is the expected return after taking an action a in a state s, and thereafter following a policy π, where

$$Q^\pi(s, a) = E_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s, a_t = a \right]. \quad (2)$$

For each MDP, there exists a set of optimal policies π^*, which share the same optimal $state\text{-}value\ function$, V^*, defined as $V^*(s) = \max_\pi V^\pi(s)$, and $action\text{-}value\ function$, Q^*, defined as $Q^*(s, a) = \max_\pi Q^\pi(s, a)$. The goal of the agent is to learn an optimal policy π^* from its interactive experience.

Most RL algorithms are designed to learn state or action value functions, e.g., an agent with Temporal Difference (TD) learning methods like Q-learning, SARSA, TD(λ) intends to learn an action value function $Q(s, a)$. For value function methods, the TD error δ_t

$$\delta_t = r_t + \gamma V(s') - V(s) \quad (3)$$

is used to update the value function as in (4), where r_t is the reward received at time step t, γ is the discounting factor, $V(s)$ and $V(s')$ are state values of current state s at time step t and next state s' at time step $t + 1$, respectively, α is the learning rate.

$$V(s) = V(s) + \alpha \delta_t. \quad (4)$$

The TD error δ_t describes how much better or worse a transition went than expected. Once the agent learned an optimal value function, it can easily obtain an optimal policy by greedily selecting actions with the optimal value function. Here, the term "greedy" means for any states the agent encountered, it selects the action with the maximum value with respect to the value function. An agent with value function methods usually learns faster since it has a lower variance in the estimates of expected return. However, value function methods need to optimize over all possible actions in every encountered state to find the optimal action, which will lead to intensive computation if the action space is continuous. Therefore, for domains with continuous action space, value function methods usually work by performing a discretization process over the action space.

In contrast with value function methods, policy search methods do not learn value functions at all; instead, they directly search the policy space to find the one receiving the most accumulated reward, which becomes an optimization problem. A policy search algorithm evaluates the performance of a candidate policy by comparing the total reward received in one or more episodes following that policy. Therefore, a performance or cost function is needed to assess the whole trajectory for each episode. There are mainly two categories of policy search approaches available, gradient methods [13]–[17] and evolutionary methods [18]–[20]. Policy search methods can work well in domains with continuous action spaces since a parameterized policy is represented. However, policy search methods usually learn very slowly because of the high variance in the estimate of the gradient.

Actor-critic—a combination of value function methods and policy gradient methods was also proposed, in which an agent learns a parameterized policy—the actor, and a parameterized value function—the critic, separately at the same time. For Actor-critic methods, the TD error δ_t in (3) is normally used to update both the policy and value function parameters. Actor-critic methods combine advantages of value function and policy search methods. The parameterized actor allows for computing with continuous action spaces and the critic speeds up the learning process by updating the parameterized policy with lower variance of gradients. Actor-critic methods usually have good convergence properties, compared to value function methods. For details about actor-critic methods, refer to the survey on actor-critic RL [11].

When an RL agent learns from evaluative feedback provided by a human teacher and not a reward function predefined by an agent designer, we take it as human-centered RL. In Section III, we will introduce the concept of human-centered RL and review algorithms and methods for learning from human evaluative feedback.

III. Human-Centered Reinforcement Learning

Human-centered RL, also known as interactive RL, is inspired by potential-based reward shaping [4]. In human-centered RL, human evaluative feedback is used to shape the agent learner [21], [22]. The objective is to facilitate the agent to learn from a human observer, especially nonexperts in agent design and programming, and speed up the agent learning at the same time. In human-centered RL, every time the agent takes an action in a state, the observing human teacher can provide evaluative feedback which tells the quality of the selected action based on the teacher's knowledge, as shown in Fig. 2. The agent then uses the evaluative feedback to update its policy. Therefore, the agent can learn how to perform a task online by interacting with a human teacher and task environment, and it is the human evaluative feedback that decides the agent's behavior. Many algorithms of human-centered RL have been developed in the past decade, and the difference between them lies at the interpretation of human feedback, e.g., numeric reward, categorical feedback strategy, and policy feedback, which will be discussed in Section III-A–III-C, respectively.

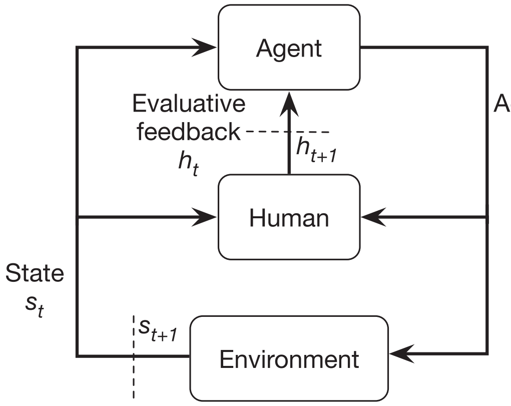


Fig. 2. Interaction in the human-centered reinforcement learning framework.

TABLE I
Human-Centered Reinforcement Learning Algorithms Categorized According to Interpretations of Human Evaluative Feedback

Interactive shaping	Thomaz and Breazeal [21],[23] Isbell *et al.* [24],[25] Tenorio *et al.* [22] Knox and Stone [26],[27] Pilarski *et al.* [28] Suay *et al.* [29] Vien and Ertel [30]
Learning from categorical feedback	Loftin *et al.* [31] ,[32]
Learning from policy feedback	Cederborg *et al.* [33] Griffith *et al.* [34] MacGlashan *et al.* [35]

Depending on the teacher's expectation of how the learner interprets evaluative feedback, there are mainly three kinds of methods developed to facilitate agents to learn from them: interactive shaping (also called learning from human reward), learning from categorical feedback strategy and learning from policy feedback, as shown in Table I. Similar to traditional RL, in interactive shaping, agents take human feedback as numeric reward values. In learning from categorical feedback strategy, agents interpret human feedback as categorical feedback strategies that depend both on the behavior the teacher is trying to teach and the teacher's training strategy. In learning from policy feedback, agents formalize the meaning of human feedback as a comment on the agent's behavior based on the expected agent policy or the policy the agent is following, and use it directly as policy feedback.

A. Interactive Shaping

In interactive shaping, a human observer provides shaping rewards to train an agent to perform a task. One of the earliest attempts to train artificial agents in this way is based on *clicker training* [36], [37]. Clicker training is a form of animal training, in which the sound of an audible device associated with a primary

TABLE II
INTERACTIVE SHAPING METHODS, CATEGORIZED ALONG TWO AXES: DISCOUNTING ON HUMAN REWARD (MYOPIC AND NONMYOPIC) AND LEARNING METHOD

	Myopic ($\gamma = 0$)	Non-Myopic ($0 < \gamma < 1$)
Value Function	Knox and Stone [26]	Knox and Stone [27] Thomaz and Breazeal [21],[23] Isbell *et al.* [24],[25] Tenorio *et al.* [22] Suan *et al.* [29] Le *et al.* [38]
Actor-Critic	Vien and Ertel [30]	Pilarski *et al.* [28]

reinforcer such as food is used as reward signal to shape the agent toward desired behaviors. Isbell *et al.* first used both reward and punishment to train an artificial agent Cobot by applying RL in an online text-based virtual chatting room [24], [25].

When learning from human reward, the agent takes the reward signal as a measure of task performance. However, since the shaping reward is provided by a human teacher and not a potential function [4], the optimality of the agent's policy in the task can not be guaranteed. Depending on whether considering the effect of an agent's action on future states or not, algorithms that learn from human reward can be mainly divided into two categories: learning from myopic human reward and learning from nonmyopic human reward. In each category, according to whether an agent learns a value function alone or both value function and policy separately at the same time, we can also group them into value function method and actor-critic method, as shown in Table II.

1) Learning From Myopic Human Reward: An agent can learn fully myopically from human reward, i.e., the discount factor γ is equal to 0. The trend of myopic in learning from human reward was identified and justified by Knox and Stone [39]. And actually nearly all approaches for learning from human reward are relatively myopic, with abnormally high rates of discounting on human reward. For example, algorithms for learning from human reward in [21], [22], [25], [29], [38] do not model human reward explicitly and use discount factors γ of 0.7, 0.75, or 0.9, where in traditional RL discount factors are almost always at or near 1 [40]. Note that, in this paper, when we say that an agent learns from myopic human reward, it means the agent learns fully myopically, i.e., the discount factor γ is 0. For methods with the discount factor γ greater than 0, we take them as learning from nonmyopical human reward.

a) Value Function Method: Value function method (also called critic method) uses TD [41] to estimate the expected return. Value function tells the expected utility of taking a given action in a given state and following the policy thereafter. The policy can be derived from the learned value function straightforwardly, e.g., with a greedy action selection strategy, which means the agent selects the action with the highest value for each state. When learning fully myopically, i.e., the discount factor is 0, the value function is equivalent to the reward function.

Knox and Stone proposed the TAMER (Training an Agent Manually via Evaluative Reinforcement) framework that learns myopically by directly modeling human reward [26]. TAMER considers an agent learning in the framework of MDP without a reward function, denoted as MDP\ R. In TAMER, a human teacher observes the agent's behavior and gives rewards based on the evaluation of its quality. The agent learns a function $\hat{R}_H(s,a) = \vec{w}^{\mathrm{T}}\Phi(s,a)$, a parameterized function approximaor that approximates the expectation of experienced human rewards, $R_H : S \times A \rightarrow \Re$, where $\vec{w} = (w_0, \ldots, w_{m-1})^{\mathrm{T}}$ is a column parameter vector, and $\Phi(\vec{x}) = (\phi_0(\vec{x}), \ldots, \phi_{m-1}(\vec{x}))^{\mathrm{T}}$ with $\phi_i(\vec{x})$ as the basis function, $i = 0, \ldots, m-1$, and m is the total number of parameters. Given a state s, the agent myopically chooses the action with the largest estimated expected reward, $\arg\max_a \hat{R}_H(s,a)$. That is to say, a TAMER agent tries to maximize the human reward caused by its immediate action while in traditional RL an agent seeks the largest discounted sum of future rewards.

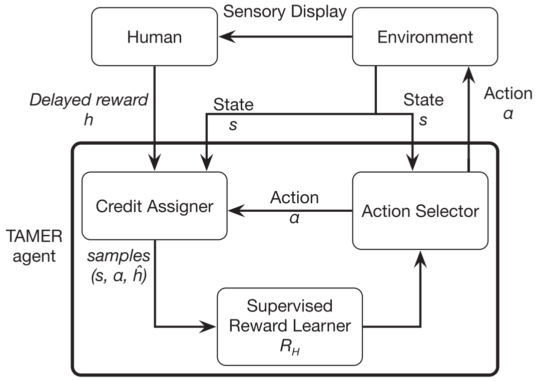


Fig. 3. Agent learning via interacting with a human teacher and the environment in the TAMER framework (reproduced from [40]).

An agent learning with TAMER includes three key modules as follows:

1) credit assignment;
2) a predictive model of human reward learned from previously experienced human rewards;
3) action selection with the learned model of human reward.

Fig. 3 depicts the interactions for agent learning in the TAMER framework.

Since it takes time for the human teacher to assess the agent's behavior and deliver her feedback, the agent is uncertain which time steps the human reward is targeting. TAMER uses a credit assignment technique to deal with the delay of human reward. Specifically, inspired by the research on the delay of human's response in visual searching tasks of different complexities [42], TAMER defines a probability density function to estimate the probability of the teacher's feedback delay. This probability density function provides the probability that the feedback occurs within any specific time interval and is used to calculate the probability (i.e., the credit) that a single reward signal is targeting at a time step. If a probability density function $f(t)$ is used to define the delay of the human reward, then at time step t, the credit for each previous time step $t-k$ is computed as

$$c_{t-k} = \int_{t-k-1}^{t-k} f(x)dx. \tag{5}$$

If the human teacher gives multiple rewards during one time step, the label h for each previous time step (state-action pair) is the sum of all credits calculated with each human reward according to (5). The TAMER agent takes the calculated label and state-action pair as a supervised learning sample to learn a human reward model–$\hat{R}_H(s,a)$. If at any time step t, the human reward label received by the agent is h, the TD error δ_t is calculated as

$$\begin{aligned}\delta_t &= h - \hat{R}_H(s,a)\\ &= h - \vec{w}^{\mathrm{T}}\Phi(s_t,a_t).\end{aligned} \tag{6}$$

Based on the gradient of least square, the parameter of $\hat{R}_H(s,a)$ is updated with incremental gradient descent.

$$\begin{aligned}\vec{w}_{t+1} &= \vec{w}_t - \alpha\nabla_{\vec{w}}\frac{1}{2}\left\{h - \hat{R}_H(s_t,a_t)\right\}^2\\ &= \vec{w}_t - \alpha\nabla_{\vec{w}}\frac{1}{2}\left\{h - \vec{w}^{\mathrm{T}}\Phi(s_t,a_t)\right\}^2\\ &= \vec{w}_t + \alpha\left\{h - \vec{w}^{\mathrm{T}}\Phi(s_t,a_t)\right\}\Phi(s_t,a_t)\\ &= \vec{w}_t + \alpha\delta_t\Phi(s_t,a_t)\end{aligned} \tag{7}$$

where α is the learning rate. A TAMER agent was shown to be able to learn faster than a traditional RL agent, but a well-tuned traditional RL agent can generally obtain a higher peak performance after many more trials [26]. TAMER was further tested and proved to be successful in many simulation domains [40], [43] such as Tetris, Mountain Car, Cart Pole, Grid World, Keepaway Soccer, Infinite Mario, and Interactive Robot Navigation.

In addition, similar to potential-based reward shaping [4], human reward can also be used to speed up agent learning from environmental reward (MDP reward). For example, the TAMER+RL framework [39], [44] allows an agent to learn from both human and environmental rewards, which can lead to a better agent performance than learning from either alone. The agent can learn sequentially first from human evaluative feedback, then environmental reward [44] and from both rewards simultaneously, which allows the human teacher to provide evaluative feedback at any time during the training process [39].

b) Actor-Critic Method: Like any value function method, the original TAMER framework only works for domains with discrete action space, Vien and Ertel extended the TAMER framework to train agents in continuous state and action domains by proposing actor-critic TAMER [30].

In actor-critic TAMER, the agent learns a human reward function, called the critic, and a parameterized policy to select actions, called the actor. A reward function, $\hat{R}_H(s) = \vec{w}^{\mathrm{T}}\Phi_r(s)$, is used, which depends only on state. The policy is represented explicitly as $\pi(s,a) = \vec{\theta}^{\mathrm{T}}\Phi_\pi(s,a)$, and separately from the reward function. The actor is used to select actions and the critic is used to evaluate the performance of the actor. The critic's evaluation provides a TD error δ_t as the gradient estimate of a specific performance measure to improve the actor by updating its parameters.

The credit of human reward for each time step, the label h, and TD error are calculated in the same way as the original TAMER framework using (5) and (6). The parameter of the learned reward function is correspondingly updated with (7). The policy $\pi(s,a)$ is updated as

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \beta\delta_t\Phi_\pi(s_t,a_t) \tag{8}$$

where β is the learning rate for the policy. Actor-critic TAMER was proved to be feasible in learning a human teacher's preferred policy in two RL benchmarking domains with continuous state and action spaces: Mountain Car and Cart Pole [30].

2) Learning From Nonmyopic Human Reward: There are certain limitations for learning fully myopically from human reward, e.g., the teacher needs to tell the correctness of behaviors in any context. Instead, learning nonmyopically could allow the teacher to communicate high-level intents [45]. Built upon this understanding, some researchers developed algorithms to learn from nonmyopical human reward.

a) Value Function Method: Knox and Stone proposed *VI-TAMER*, a model-based RL method, which facilitates an agent to learn from nonmyopic human reward [27], [46]. In *VI-TAMER*, an agent learns from discounted human reward and models the human reward at the same time.

VI-TAMER is composed of two established algorithms running in parallel: 1) TAMER learns predictive models of human reward with the agent's experienced state-action pairs and human reward provided by the human teacher, and 2) value iteration [1] learns a state value function with the predictive model of human reward, as

$$V(s) \leftarrow \max_a[\hat{R}_H(s,a) + \gamma\sum_{s'\in S}T(s,a,s')V(s'))] \tag{9}$$

where $T(s,a,s')$ is the transition function which tells the probability of the agent selecting an action a in a state s and transitioning to next state s', $V(s')$ is the state value of next state s'. During each time step, value iteration updates values of all possible states with the most recent human reward function learned from TAMER, as in (9). Therefore, value iteration updates the value function much more often than the agent's experienced time steps. The agent can quickly adapt its value function to the most recent reward function changed from TAMER.

In contrast with the typical value iteration, the TD error for updating the state value function $V(s)$ is calculated with the modeled human reward function:

$$\delta_t = \hat{R}_H(s,a) + \gamma V(s') - V(s). \tag{10}$$

As is typical for value iteration, the agent chooses actions with the largest state-action value of all potential actions in next state s' according to the learned value function and human reward function.

$$a \leftarrow \arg\max_a[\hat{R}_H(s,a) + \gamma\sum_{s'\in S}T(s,a,s')V(s'))]. \tag{11}$$

When the discount factor γ is 0, the learned value function $V(s)$ in VI-TAMER is equivalent to the learned reward function $\hat{R}_H(s,a)$ in the TAMER framework. Therefore, the original TAMER can be regarded as a special case of VI-TAMER. VI-TAMER was tested in both episodic and continuing Grid World tasks with 30 states [27]. Their results showed that in episodic tasks, agents should learn myopically from human reward. Moreover, they showed that in continuing tasks, agents learning nonmyopically from human reward are more robust to environmental changes and more able to act appropriately in previously unexperienced states.

In addition, different from VI-TAMER, model-free RL methods can also be used for agents learning from nonmyopic human reward. For example, a Q-value function [47] can also be learned directly from human reward without modeling it [21], [29]. In this case, human rewards are taken in the same way as environmental rewards in traditional RL. Moreover, an agent can also learn from nonmyopic human reward and environmental reward at the same time. For example, Thomaz and Breazeal [21], [23] implemented an interface with a *Q-learning* [47] agent which aims to maximize its total discounted sum of human reward and environmental reward.

b) Actor-Critic Method: Similar to actor-critic TAMER, Pilarski *et al.* [28] proposed a continuous action model-free RL algorithm that learns an optimal control policy using only human reward. Their algorithm learns from discounted human reward as in nonmyopic TAMER [27], but works with continuous state and action spaces.

However, unlike VI-TAMER, their algorithm does not model human reward signals. Instead, they treat human rewards in the same way as environmental rewards in traditional RL. Therefore, in their algorithm, the agent uses the original human reward h and tries to learn a value function $V(s)$ and policy π to receive the most discounted accumulated human reward:

$$V(s) \leftarrow \max_a [h + \gamma \sum_{s' \in S} T(s, a, s')V(s'))]. \tag{12}$$

Since human reward is used in the same way as environmental MDP reward in traditional RL, the TD error is calculated as

$$\delta_t = h + \gamma V(s') - V(s). \tag{13}$$

Then for each time step, δ_t is used to update the value function $V(s)$ and policy π as in actor-critic RL with (7) and (8). With actor-critic RL from nonmyopic human reward, an agent was tested and shown to be able to learn a control policy for a simulated robotic arm, indicating the potential use for the myoelectric control of powered prostheses [28].

B. Learning From Categorical Feedback Strategy

While the work in Section III-A interprets human feedback as a numeric reward, Loftin *et al.* [31], [32] interpret human feedback as categorical feedback strategies that depend on the behavior the teacher is trying to teach as well as the teacher's training strategy. They assume that human teachers provide feedback with different strategies, and for strategy-aware learners, the lack of feedback can be as informative as explicit feedback.

Specifically, motivated by behaviorism and animal training, the way that teachers provide feedback can be divided into four categories: positive reward (R+), negative reward (R−), positive punishment (P+), and negative punishment (P−). Here, reward means a stimulus that would increase the frequency of an associated behavior, while punishment would be a stimulus that decreases the frequency of a behavior. Positive refers to adding a stimulus and negative refers to removing a stimulus. Both positive and negative reward encourage an associated behavior, while both positive and negative punishment discourage an associated behavior. In particular, it means that different teachers can use different ways to provide feedback, even when they are teaching the same behavior. For example, when the learner takes a correct action, one teacher might provide an explicit positive reward (R+), while another might provide no response at all (R−).

Based on the way that teachers provided explicit feedback in their studies, Loftin *et al.* divide the teachers' strategies into four groups: reward-focused strategy, punishment-focused strategy, balanced strategy, and inactive strategy, which correspond to the so-called operant conditioning paradigms in behaviorism [48]. For example, with a reward-focused strategy, a teacher provides explicit rewards for an agent's correct actions and no response for incorrect actions, which roughly corresponds to a R+/P− paradigm in behaviorism. With a punishment-focused strategy, a teacher would provide no feedback for correct actions and explicit punishments for incorrect ones, which would correspond to a R−/P+ paradigm. With an inactive strategy, a teacher rarely gives explicit reward or punishment feedback, which corresponds to a R−/P− paradigm. Under a balanced feedback strategy, both explicit reward and explicit punishment are used, which corresponds to a R+/P+ paradigm. Therefore, under a reward-focused strategy, the lack of feedback can be interpreted as implicit negative feedback, while under a punishment-focused strategy, it can be interpreted as implicitly positive.

Loftin *et al.* developed a probabilistic model of evaluative feedback to capture teachers' categorical feedback strategies, and use this model to build learning algorithms with probabilistic inference to identify target behaviors based on the teacher's strategy. Specifically, the learning problem is modeled as a task with discrete observations of the environment and discrete actions to be taken. The target behavior to be trained is denoted as a policy π, mapping from observations to actions. The training is divided into discrete episodes. The agent in the environment first observes the environment and takes an action. Then the teacher determines if the action taken by the agent is consistent with the target policy π^* for the current observation, with probability of error ϵ, and may give explicit reward or punishment, or no feedback at all. They define the parameters $\mu^+ \in [0, 1]$ and $\mu^- \in [0, 1]$ as the probability of giving no feedback for the agent's correct and incorrect actions, representing the teacher's strategy.

Therefore, if the agent takes a correct action, the teacher will give an explicit reward with probability $(1 - \epsilon)(1 - \mu^+)$, an explicit punishment with probability $\epsilon(1 - \mu^-)$, and no feedback with probability $(1 - \epsilon)\mu^+ + \epsilon\mu^-$. If the agent takes an incorrect action, the teacher will give an explicit reward with probability $\epsilon(1 - \mu^+)$, an explicit punishment with probability $(1 - \epsilon)(1 - \mu^-)$, and no feedback with probability $\epsilon\mu^+ + (1 - \epsilon)\mu^-$. Then for any time step t, the agent has a distribution of the teacher's feedback $f_t, p(f_t|o_t, a_t, \pi^*(o_t) = a')$, conditioned on the observation o_t, action a_t, and the teacher's target policy π^*.

Based on the previous probabilistic model with which the teacher provides feedback, they developed the strategy-aware

Bayesian learning (SABL) algorithm, which computes a maximum likelihood estimate of the teacher's target policy π^* online given the feedback that the user has provided; that is, it computes

$$\text{argmax}_{\pi} p(h_{1,\ldots,t}|\pi^* = \pi) \tag{14}$$

where h_t is the training history consisted of actions, observations, and feedback. Specifically, at any time step t, the agent observes the state of the environment, then it selects and executes an action with policy $P(o_t, a')$. The agent will receive feedback from the human teacher and interpret the lack of feedback according to the teacher's specified strategy. Based on the received feedback, it updates the policy $P(o_t, a')$ with the previous distribution of feedback $p(f_t|o_t, a_t, \pi^*(o_t) = a')$:

$$P(o_t, a') \leftarrow p(f_t|o_t, a_t, \pi^*(o_t) = a')P(o_t, a'). \tag{15}$$

If a teacher provides multiple feedback, SABL only considers the most recent one, allowing a teacher to correct a previous mistaken feedback.

SABL assumes the teacher's strategy is known before learning, i.e., parameters μ^+ and μ^- are available, though in practice the teacher's strategy is mostly unknown and the teacher even might change the strategy during training. Therefore, under SABL's probabilistic model, they treat the parameters μ^+ and μ^- as unknown and use them to represent the teacher's strategy. Then they compute the likelihood of a possible target policy π by marginalizing over possible strategies. I-SABL (Inferring-SABL), was developed to find a maximum likelihood estimate of the target policy given the training history. Specifically, Expectation Maximization [49] algorithm was used to compute a maximum likelihood estimate of the target policy by treating the unknown μ^+ and μ^- parameters as continuous, hidden variables ranging from 0 to 1.

$$\text{argmax}_{\pi} \sum_{s \in S} p(h_{1,\ldots,t}, s|\pi^* = \pi) \tag{16}$$

where S is the set of possible training strategies represented with μ^+ and μ^- values, $p(s)$ is uniform for all $s \in S$, and $h_{1,\ldots,t}$ is the training history up to the current time t. In a contextual bandit domain where the action selected by an agent based on the observation of the environmental state can only affect the probability of immediate feedback rather than subsequent states, agents with SABL and I-SABL were shown to be able to learn better than methods that learn from numeric rewards [32].

C. Learning From Policy Feedback

1) Policy Shaping: Instead of converting feedback signals into evaluative rewards, policy shaping [34] formulates human evaluative feedback as policy labels and uses these labels directly to infer what the human teacher believes is the optimality of the labeled action in a state.

Policy shaping assumes that when a human teacher provides feedback she knows the right answer, but noise in the feedback channel introduces inconsistencies between what the human teacher intends to communicate and what the agent observes. Thus, the probability of human feedback consistent with the optimal policy is denoted as C, where $0 < C < 1$. That is to say, when an agent takes a correct (or incorrect) action, the human teacher is assumed to give a positive (or negative) feedback with probability C.

It is reasonable to assume that a human teacher might believe a number of different actions to be optimal for a given state. In a state s, the probability that an action is optimal is independent of what feedback was provided to the other actions. Therefore, the probability of an action a being optimal in a given state s, $Pr_c(a)$, can be computed with only the human feedback associated with it.

$$Pr_c(a) = \frac{C^{\Delta_{s,a}}}{C^{\Delta_{s,a}} + (1-C)^{\Delta_{s,a}}} \tag{17}$$

where $\Delta_{s,a}$ is the number of positive minus negative human feedback labels in the training data for the action a in state s. $Pr_c(a)$ is the probability of taking action a in state s according to human feedback policy π_F.

The estimated policy π_F can be used to affect the action selection in agent learning with a traditional RL algorithm, like TAMER+RL [39], [44]. Specifically, π_F can be combined with the learned policy from a traditional RL algorithm by multiplying probability distributions together with the Bayes optimal method [50]. For example, Griffith *et al.* [34] combined the estimated feedback policy $Pr_c(a)$ from human feedback with the learned policy $Pr_q(a)$ from Bayesian Q-learning [51] which learns a distribution of each Q value and uses it to estimate the probability that each action a is optimal in a state s. In addition, Cederbog *et al.* [33] used a Q-learning agent with Boltzmann exploration to compute $Pr_q(a)$.

$$Pr_q(a) = \frac{e^{Q(s,a)/\tau}}{\sum_{a'} e^{Q(s,a')/\tau}} \tag{18}$$

where τ is a temperature constant. Then, the probability of selecting action a in a state s with policy shaping is

$$Pr(a) = \frac{Pr_q(a)Pr_c(a)}{\sum_{a' \in A} Pr_q(a')Pr_c(a')}. \tag{19}$$

Griffith *et al.* compared the learning performance of policy shaping to that of TAMER+RL and potentially reward shaping in a series of experiments with a simulated human teacher. Their results showed that policy shaping has similar performance to these state-of-the-art methods, but is more robust to infrequent and inconsistent feedback [34]. In addition, Cederbog *et al.* [33] extended their work by evaluating policy shaping with real human teachers and showed that policy shaping is suitable for human generated data and participants in the experiment even outperform the simulated teacher.

2) COACH: MacGlashan *et al.* [35] pointed out that all interactive shaping, learning from categorical feedback strategy and even policy shaping algorithms interpret human evaluative feedback as comments on agent's behaviors based on the expected agent's policy and independent of the agent's current policy. Evidenced by the decreasing feedback rates in previous studies, they claimed that human evaluative feedback should be interpreted as policy feedback depending on the agent's current policy, and proposed an actor-critic algorithm—Convergent Actor-Critic by Humans (COACH), to learn from human feedback.

MacGlashan *et al.* further claimed that current policy-dependent feedback can afford three desirable human training strategies as follows.

1) Gradually decrease positive feedback for good actions as the agent successfully learned those actions.
2) Vary the magnitude of feedback with respect to the degree of improvement or deterioration in behavior.
3) Provide positive feedback for suboptimal actions that improve the agent's behavior and then negative feedback after the improvement has been made.

They defined the model of human evaluative feedback as the advantage function [52] to capture those three training strategies. The advantage function describes how much better or worse an action selection is compared to the agent's performance following policy π:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s). \tag{20}$$

COACH is based on the insight that actor-critic algorithms update the agent's policy parameters with TD error and TD error is simply an unbiased estimate of the advantage function [53]. Therefore, if human evaluative feedback accurately approximates the advantage function, then the agent learning can be performed with an actor-critic algorithm that sets the TD error δ_t directly to the human feedback value:

$$\delta_t = h. \tag{21}$$

In COACH, a human teacher can give multiple human feedbacks. The feedback signal h represents the sum of human feedback values given in the time between the action execution and the resulting state being observed. COACH was shown to be able to successfully learn five navigational behaviors on a TurtleBot robot fast. In comparison with TAMER, COACH can learn to combine subbehaviors with compositional training methods, while TAMER can only learn those behaviors with a flat training methodology [35].

In summary, all human-centered RL methods can facilitate an agent to learn a human preferred policy online based on the received evaluative feedback from the human teacher, while interacting with the environment at the same time. Both *learning from categorical feedback strategy* and *policy shaping* use a model of the feedback distribution to estimate a posterior distribution over the trainer's policy. The interpretation of silence in *policy shaping* is similar to *learning from categorical feedback strategy*, in which information from cases where no feedback is provided is inferred and used to estimate the human teacher's feedback policy. However, both *learning from categorical feedback strategy* and *interactive shaping* can learn and be tested exclusively from human feedback, while policy shaping is only tested while running together with other traditional RL algorithms. Therefore, the effectiveness of *policy shaping* with only human evaluative feedback and tasks with continuous state and action remain to be further explored. In COACH, for the human teacher to provide feedback in the form of advantage function, they would need to observe the agent's behavior and estimate its current policy. In practice, it might be harder for the human teacher to estimate the agent's current policy than the expected one.

IV. Improving the Efficiency of Human-Centered Reinforcement Learning

Like other interactive learning methods, successful learning from human evaluative feedback depends on the efficiency of the interaction between the human teacher and the agent learner. Research shows that a bidirectional interaction should be built between the agent and human teacher [54]: the teacher needs to provide feedback to teach the agent, and the agent should also give explicit feedback to the teacher on its learning state. Therefore, to allow nonexpert users to effectively teach agents how to perform tasks, researchers studied what kinds of interfaces should be designed to allow agents to learn interactively from human teachers in a natural way. Specifically, they investigated what kinds of information or feedback the human teacher and agent could provide to improve the agent's learning efficiency.

A. Using Human Feedback

From the human teacher's point of view, Thomaz and Breazeal [21], [23] found that a human teacher might have many different kinds of communicating intents with one interaction channel, e.g., feedback, guidance, motivation, etc., and the agent could take better advantage of messages from them. They investigated whether allowing the teacher to give guidance can improve an agent's learning from human reward. Specifically, they implemented an interface with a tabular *Q-learning* [47] agent. An interaction channel is introduced in the interface to allow the human teacher to give the agent evaluative feedback. Moreover, the channel allows the human teacher to give guidance—action advice to bias action selection by narrowing down the action space. Their results show that human's guidance can lead the agent to explore more efficiently, and the agent's learning performance was improved as a result.

B. Using Agent Feedback

From the agent's point of view, the transparency and responsiveness of an agent's behavior can improve the interaction efficiency. Thomaz and Breazeal [55] investigated while learning with human reward and guidance, how an agent can take advantage of an agent's gaze behavior to improve its learning performance in a virtual kitchen environment. Specifically, the agent's gaze behavior is used to communicate next action that the agent is going to follow to a human teacher. The number of gazes is determined by the number of actions with highest action values within some bound, thus more gazings signal a higher level of uncertainty. Their results show that the agent's gaze behavior as a transparent act can help the teacher understand when the agent does (and does not) need guidance.

To build a joint activity between the human teacher and robot learner and further engage the teacher, in the transparent learning mechanism, robots are allowed to use facial expressions and body languages, such as gaze direction, eye blinks, shifts in gaze, and shifts in the body gesture between actions, shrugging gestures, etc., to express its learning state and solicit feedback from the human teacher [56], [57]. Meanwhile, the human teacher can influence the learning process through attention direction, action suggestions, labeling goal states, and evaluative feedback [58].

In addition, with TAMER as a foundation, Knox *et al.* [59] examined how human teachers respond to changes in their perception of the agent and its behavior. They found that when the quality of the agent's behavior is reduced whenever the rate of human feedback decreases, the agent can solicit more feedback from the human teacher but with lower performance. Li *et al.* also studied what information the agent should share with the teacher to express how well the agent's learning is going and its effect on the teaching process and agent's learning. They found that the information the agent shares with the human teacher has a great effect on the quantity of human provided feedback and the agent's learning performance [54], [60], [61].

V. Applications

In this section, we provide references to work on applying human-centered RL algorithms in some robotic domains. Though the list of applications is not exhaustive, it gives an impression of the possibility of robot learning from human evaluative feedback and describes the challenges and lessons learned in applying human-centered RL algorithm to embodied robots.

In robot learning, early successful application of human-centered RL algorithm was shown on a real autonomous mobile robot learning to perform navigation tasks in a simulated environment [22]. Specifically, Tenorio *et al.* [22] proposed to allow the robot to learn from both explicit demonstrations and human rewards as well as environmental rewards. In their work, both the demonstration and human reward were provided via verbal commands and human rewards were used as dynamic shaping rewards to accelerate robot learning from environmental rewards. Even though human rewards are noisy, faster convergence was achieved compared to traditional RL.

Suay and Chernova first applied interactive shaping on a real-world robotic system with an Aldebaran humanoid Nao robot in an object sorting domain [29]. Specifically, in the domain, the robot needs to learn to pick up and place magnetic objects in one of two cups on a table. The human teacher observes the robot's action via a web camera and provides evaluative rewards to the robot through a graphical user interface. In their work, the robot learns only from the human reward using a Q-learning algorithm with discrete state and action spaces. They also tested robot learning from both human reward and guidance. They claimed that challenges such as the time of executing an action, extension to complex domains with continuous state and action spaces, and how to automate feature selection for state representation in large domains need to be further studied when applying interactive shaping in real-world environments.

Knox *et al.* [45] first applied the interactive RL algorithm that deals with delay of human rewards on a Mobile Dexterous Social robot "Nexi," as shown in Fig. 4. They implemented the TAMER framework for learning from numeric human feedback to train "Nexi" to learn behaviors in five different interactive navigational tasks. This is the first time to train multiple behaviors on an embodied robot purely with human reward. Like in [29], they claimed that the execution time of the robot's action is an issue likely to occur with any physically embodied robot trained with human reward, which can cause mismatches between the state-action pair currently occurring and what the teacher believes to be occurring. Thus, human reward might be misattributed to the wrong action. Therefore, the duration of robot's action needs to be carefully selected to make sure the execution of the action visible and can be targeted by the human teacher.

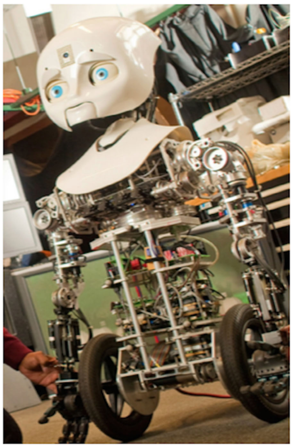


Fig. 4. Interactive shaping method, TAMER, applied to a wheeled robot Nexi which learns to perform five different interactive navigational tasks in [45].

MacGlashan *et al.* [35] applied the COACH algorithm that learns from policy-dependent feedback on a physical TurtleBot robot. The TurtleBot is a mobile base with two degrees of freedom that senses the world with a Kinect camera. They used COACH to train the robot to perform five different navigational behaviors involving a pink ball and cylinder with an orange top. The same parameter selections for the algorithm were used, and they compared the robot learning to the TAMER framework. They showed that a robot with COACH was able to successfully learn all five behaviors with both flat and compositional training methods. Each of these behaviors was trained in less than 2 min. The robot with TAMER also successfully learned the five behaviors but failed to learn the compositionally trained behaviors.

In addition, Pilarski *et al.* [28] tested a continuous action actor-critic interactive shaping algorithm that learns an optimal control policy for a simulated upper-arm robotic prosthesis from only discounted human reward. In their experiment, a human user teaches the robot arm to learn a control policy that outputs joint velocity commands to match three activities—reaching, retracted, and relaxed. The effectiveness of their method on real robotic prosthesis needs to be further investigated. However, teaching by human-generated reward remains to be a promising technique for improving the effectiveness of myolectric and EEG-based Brain Machine Interfaces, which are being developed to enable handicapped users to control various robotic devices.

Though the evaluation of human-centered RL algorithms on physically embodied robot is limited with respect to only the authors training the robot, it would be straightforward to extend to nontechnical users. Further work with numerous nonexpert human teachers and other task domains will be critical to establishing the generality of the application of learning from human evaluative feedback on real-world robotic systems.

VI. Discussion and Outlook

Though many human-centered RL algorithms have been proposed, it is still a young research direction. Based on recent advances in artificial intelligence, there are many potential problems remaining to be further explored.

A. Combine With Other Interactive Learning Methods

Besides learning from human evaluative feedback, there are several different ways developed for nontechnical people to teach autonomous agents naturally, e.g., demonstrations, instructions, or advice. In the real-world applications, people might prefer to use more than one way to teach agents how to perform tasks. A Wizard of Oz study [62], [63] showed that human rewards are usually used by teachers to fine tune the agent's behavior learned from demonstration. To facilitate the human teacher to train an agent with both demonstrations and human rewards, Li *et al.* [64] proposed an IRL-TAMER framework that allows an agent to learn from demonstration via inverse RL [65] first and then from human reward via TAMER. They studied the effect of demonstrations on an agent's learning from human reward and found that demonstrations can decrease the quantity of human evaluative feedback needed for the agent to learn an optimal policy, especially the negative one. That is to say, demonstration can reduce the number of incorrect actions during the learning process with trial and error. However, methods that allow the interchangeability of demonstrations and human rewards, or even in combination with instruction or advice need to be further investigated in the future.

B. Learning From Implicit Human Reward

From a human perspective, as time progresses in the training process, human teachers might get tired of giving *explicit evaluative feedback*. In fact, several studies with the TAMER framework have shown that teachers give copious feedback in the early training stage but very sparsely thereafter [59], [66]. In the real world, people often consciously or subconsciously use other implicit feedback such as facial expressions to encourage or discourage specific behaviors they like, e.g., smile indicates a good behavior and frown means a bad one [67], in addition to giving explicit encouragement or punishment. Therefore, it would be useful to investigate how an agent can learn from implicit feedback such as facial expressions and even how an agent can learn from both implicit and explicit evaluative feedback.

Broekens examined how a robot can learn from emotional facial expressions by taking them as social human rewards [68]. Though only fear, happy, and neutral emotions were taken as prototypes of facial expressions and mapped to predefined numeric reward values and nine stickers on the face were used to help recognize them, their results showed that facial expressions can facilitate robot learning significantly faster compared to a robot trained without them.

In addition, Li *et al.* [69], [70] recorded the facial expressions of 498 trainers during training and built mapping between facial expression and explicit human reward. They studied how the agent can learn from facial expressions via taking them as implicit human reward without taking environmental reward into account. Their results showed that an agent can learn from facial expressions but not as well as from explicit feedback.

Recently, Veeriah *et al.* [71] proposed a method—face valuing, with which an agent can adapt to the user's preference by learning from the human user's facial expressions. Though their experiments were conducted with a single well-trained user and the user was asked to give clear clues to express pleasure or displeasure of the agent's action, their preliminary results suggest that an agent can quickly adapt to a user's changing preference by learning a value function that maps facial features extracted from a camera image to expected future reward. Moreover, they showed that learning with facial expressions can reduce the amount of explicit feedback required to complete a grip selection task.

C. Human-Centered Deep Reinforcement Learning

All human-centered RL algorithms and their applications were tested in a simple task domain or with manual selection of the appropriate state features as state representation. Therefore, human-centered RL has been limited to simple domains or those in which useful features can be handcrafted. Manually selecting state representation is time consuming and may yield suboptimal results if not done correctly. Moreover, in many social settings, it is difficult or even impossible to handcraft features to be fed into the algorithm. How to automate feature selection for state representation in large domains remains to be a challenge especially when applying human-centered RL in real-world environments.

Inspired by the human brain, researchers from Google DeepMind developed a *deep RL* agent, which can learn from raw pixels and game score to play 49 Atari games and adapt to different tasks with the same algorithm [72], [73]. *Deep RL* is a combination of deep learning [74] and RL, and can be used to design agents learning the features and how to perform the task at the same time. In deep RL, a deep neural network was learned to approximate a Q-value function in Q-learning, termed Deep Q-network. However, like standard RL, it cannot be applied to real-world agents that learn from human beings, since the optimal behavior is predefined via a reward function (together with the transition of states). Therefore, how autonomous agents learn from human evaluative feedback might also need to be rethought in light of these findings. Recently, Christiano *et al.* [75] tried to use deep RL algorithm to learn from comparisons of video clips between 1 and 2 s provided by a nonexpert human user. However, in their work trajectory segment comparisons are used as human feedback, rather than evaluative feedback on the agent's actions.

D. Deep Understanding of Human Evaluative Feedback

Teaching with human evaluative feedback is a complex behavior and process, compared to agent learning from environmental feedback in traditional RL. Thomaz and Breazeal

[21] pointed out that people want to direct the agent's attention to guide the exploration process in the teaching process, and they have a positive bias in their rewarding behaviors and even adapt their teaching strategy as they develop a mental model of how the agent learns. Therefore, we need a deep understanding of human teaching behavior before developing methods to facilitate agents to effectively learn from evaluative feedback.

Ho *et al.* [76] tried to understand how human evaluative feedback is used to shape other humans' behavior and how it is different from environmental reward and punishment, i.e., how special human evaluative feedback is. Similar to learning from categorical feedback strategy, they analyzed the logic of human evaluative feedback and claimed that adoptive learning mechanism is favored in the case of learning from evaluative feedback provided by a social partner. In adoptive learning mechanism, evaluative feedback is treated as communicating information about the value of an action rather than as a form of reward to be maximized. They assume teachers have their own value representations and reward specification, and agents can adopt the mental structures of teachers. They claimed that adoptive learning mechanism can remove the positive reward cycle problem observed in [8], [27], since it does not maximize the human reward. However, how to specify hierarchical and complex reward function from human evaluative feedback still remains to be further explored [12].

VII. Conclusion

In this paper, we have surveyed most recent work on human-centered RL research, a young and not fully understood subdiscipline of RL. We provided backgrounds in RL and reviewed human-centered RL algorithms based on the interpretations of human evaluative feedback. Methods for improving the efficiency of agents learning from human evaluative feedback and some implemented robotic systems are also surveyed. There is still much work to be done in this area and many interesting open questions remain to be solved. Experimental studies with social assistive robotics and the development of RL in the real world will inevitably shed further light onto this topic.

References

[1] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[2] A. G. Barto, R. S. Sutton, and C. J. C. H. Watkins, "Learning and sequential decision making (technical report)," in *Learning and Computational Neuroscience*. Cambridge, MA, USA: MIT Press, 1989, pp. 539–602.

[3] J. Kober and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, 2014.

[4] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. Int. Conf. Mach. Learn.*, 1999, pp. 278–287.

[5] B. Marthi, "Automatic shaping and decomposition of reward functions," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 601–608.

[6] M. Grzes and D. Kudenko, "Plan-based reward shaping for reinforcement learning," in *Proc. 4th Int. IEEE Conf. Intell. Syst.*, vol. 2, 2008, pp. 10–22.

[7] A. Harutyunyan, S. Devlin, P. Vrancx, and A. Nowé, "Expressing arbitrary reward functions as potential-based advice," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 2652–2658.

[8] M. K. Ho, M. L. Littman, F. Cushman, and J. L. Austerweil, "Teaching with rewards and punishments: Reinforcement or communication," in *Proc. 37th Annu. Meeting Cogn. Sci. Soc.*, 2015, pp. 920–925.

[9] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[10] P. Dayan and Y. Niv, "Reinforcement learning: The good, the bad and the ugly," *Current Opinion Neurobiol.*, vol. 18, no. 2, pp. 185–196, 2008.

[11] I. Grondman, L. Busoniu, G. A. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern., Part C (Appl. Rev.)*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.

[12] M. L. Littman, "Reinforcement learning improves behaviour from evaluative feedback," *Nature*, vol. 521, no. 7553, pp. 445–451, 2015.

[13] R. S. Sutton *et al.*, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 1057–1063.

[14] A. Y. Ng and M. Jordan, "Pegasus: A policy search method for large MDPs and POMDPs," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, 2000, pp. 406–415.

[15] L. Peshkin, K.-E. Kim, N. Meuleau, and L. P. Kaelbling, "Learning to cooperate via policy search," in *Proc. 16th Conf. Uncertainty Artif. Intell.*, 2000, pp. 489–496.

[16] N. Kohl and P. Stone, "Policy gradient reinforcement learning for fast quadrupedal locomotion," in *Proc. 21st IEEE Int. Conf. Robot. Autom.*, vol. 3, 2004, pp. 2619–2624.

[17] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 465–472.

[18] D. E. Moriarty, A. C. Schultz, and J. J. Grefenstette, "Evolutionary algorithms for reinforcement learning," *J. Artif. Intell. Res.*, vol. 11, pp. 241–276, 1999.

[19] X. Yao, "Evolving artificial neural networks," *Proc. IEEE*, vol. 87, no. 9, pp. 1423–1447, Sep. 1999.

[20] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies," *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, 2002.

[21] A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artif. Intell.*, vol. 172, no. 6, pp. 716–737, 2008.

[22] A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villaseñor-Pineda, "Dynamic reward shaping: Training a robot by voice," in *Proc. Ibero-Amer. Conf. Artif. Intell.*, 2010, pp. 483–492.

[23] A. L. Thomaz and C. Breazeal, "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance," in *Proc. 20th AAAI Conf. Artif. Intell.*, vol. 6, 2006, pp. 1000–1005.

[24] C. Isbell, C. R. Shelton, M. Kearns, S. Singh, and P. Stone, "A social reinforcement learning agent," in *Proc. 5th Int. Conf. Auton. Agents*, 2001, pp. 377–384.

[25] C. L. Isbell Jr, M. Kearns, S. Singh, C. R. Shelton, P. Stone, and D. Kormann, "Cobot in lambdamoo: An adaptive social statistics agent," *Auton. Agents Multi-Agent Syst.*, vol. 13, no. 3, pp. 327–354, 2006.

[26] W. B. Knox and P. Stone, "Interactively shaping agents via human reinforcement: The TAMER framework," in *Proc. 5th Int. Conf. Knowl. Capture*, 2009, pp. 9–16.

[27] W. B. Knox and P. Stone, "Framing reinforcement learning from human reward: Reward positivity, temporal discounting, episodicity, and performance," *Artif. Intell.*, vol. 225, pp. 24–50, 2015.

[28] P. M. Pilarski, M. R. Dawson, T. Degris, F. Fahimi, J. P. Carey, and R. S. Sutton, "Online human training of a myoelectric prosthesis controller via actor-critic reinforcement learning," in *Proc. 12th Int. Conf. Rehabil. Robot.*, 2011, pp. 1–7.

[29] H. B. Suay and S. Chernova, "Effect of human guidance and state space size on interactive reinforcement learning," in *Proc. IEEE Int. Symp. Robot Human Interact. Commun.*, 2011, pp. 1–6.

[30] N. A. Vien and W. Ertel, "Reinforcement learning combined with human feedback in continuous state and action spaces," in *Proc. IEEE Int. Conf. Develop. Learn. Epigenetic Robot.*, 2012, pp. 1–6.

[31] R. Loftin *et al.*, "A strategy-aware technique for learning behaviors from discrete human feedback," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 937–943.

[32] R. Loftin *et al.*, "Learning behaviors via human-delivered discrete feedback: Modeling implicit feedback strategies to speed up learning," *Auton. Agents Multi-Agent Syst.*, vol. 30, no. 1, pp. 30–59, 2016.

[33] T. Cederborg, I. Grover, C. L. Isbell, and A. L. Thomaz, "Policy shaping with human teachers," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 3366–3372.

[34] S. Griffith, K. Subramanian, J. Scholz, C. Isbell, and A. L. Thomaz, "Policy shaping: Integrating human feedback with reinforcement learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 2625–2633.

[35] J. MacGlashan *et al.*, "Interactive learning from policy-dependent human feedback," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 2285–2294.

[36] B. Blumberg, M. Downie, Y. Ivanov, M. Berlin, M. P. Johnson, and B. Tomlinson, "Integrated learning for interactive synthetic characters," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 417–426, 2002.

[37] F. Kaplan, P.-Y. Oudeyer, E. Kubinyi, and A. Miklósi, "Robotic clicker training," *Robot. Auton. Syst.*, vol. 38, no. 3, pp. 197–206, 2002.

[38] A. Len, E. F. Morales, L. Altamirano, and J. R. Ruiz, "Teaching a robot to perform task through imitation and on-line feedback," in *Proc. Iberoamer. Congr. Pattern Recognit.*, 2011, pp. 549–556.

[39] W. B. Knox and P. Stone, "Reinforcement learning from simultaneous human and MDP reward," in *Proc. 11th Int. Conf. Auton. Agents Multiagent Syst., Int. Found. Auton. Agents Multiagent Syst.*, 2012, pp. 475–482.

[40] W. B. Knox, "Learning from human-generated reward," Ph.D. thesis, Dept. Comput. Sci., Univ. Texas, Austin, TX, USA, 2012.

[41] G. Tesauro, "Temporal difference learning and TD-gammon," *Commun. ACM*, vol. 38, no. 3, pp. 58–68, 1995.

[42] W. E. Hockley, "Analysis of response time distributions in the study of cognitive processes," *J. Exp. Psychol., Learn., Memory, Cognit.*, vol. 10, no. 4, pp. 598–615, 1984.

[43] G. Li, "Socially intelligent autonomous agents that learn from human reward," Ph.D. thesis, Informat. Inst., Univ. Amsterdam, Amsterdam, The Netherlands, 2016.

[44] W. B. Knox and P. Stone, "Combining manual feedback with subsequent MDP reward signals for reinforcement learning," in *Proc. 9th Int. Conf. Auton. Agents Multiagent Syst.*, 2010, pp. 5–12.

[45] W. B. Knox, P. Stone, and C. Breazeal, "Training a robot via human feedback: A case study," in *Proc. Int. Conf. Social Robot.*, 2013, pp. 460–470.

[46] W. B. Knox and P. Stone, "Learning non-myopically from human-generated reward," in *Proc. Int. Conf. Intell. User Interfaces*, 2013, pp. 191–202.

[47] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, no. 3–4, pp. 279–292, 1992.

[48] B. F. Skinner, *Science and Human Behavior*. New York, NY, USA: Simon and Schuster, 1953.

[49] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc., Ser. B (Methodolog.)*, vol. 39, no. 1, pp. 1–22, 1977.

[50] C. Bailer-Jones and K. Smith, "Combining probabilities," Data Process. Anal. Consortium, Max Planck Inst. for Astronomy, Heidelberg, Germany, Tech. Rep. GAIA-C8-TN-MPIA-CBJ-053, 2011.

[51] R. Dearden, N. Friedman, and S. Russell, "Bayesian Q-learning," in *Proc. AAAI/IAAI Conf. Artif. Intell.*, 1998, pp. 761–768.

[52] L. Baird *et al.*, "Residual algorithms: Reinforcement learning with function approximation," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 30–37.

[53] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor–critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[54] G. Li, S. Whiteson, W. B. Knox, and H. Hung, "Using informative behavior to increase engagement while learning from human reward," *Auton. Agents Multi-Agent Syst.*, vol. 30, no. 5, pp. 826–848, 2016.

[55] A. L. Thomaz and C. Breazeal, "Transparency and socially guided machine learning," in *Proc. 5th Int. Conf. Develop. Learn.*, 2006, pp. 1–6.

[56] C. Breazeal, C. D. Kidd, A. L. Thomaz, G. Hoffman, and M. Berlin, "Effects of nonverbal communication on efficiency and robustness in human-robot teamwork," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2005, pp. 708–713.

[57] A. L. Thomaz, G. Hoffman, and C. Breazeal, "Real-time interactive reinforcement learning for robots," in *Proc. AAAI Workshop Human Comprehensible Mach. Learn.*, 2005, pp. 1–5.

[58] C. Breazeal and A. L. Thomaz, "Learning from human teachers with socially guided exploration," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2008, pp. 3539–3544.

[59] W. B. Knox, B. D. Glass, B. C. Love, W. T. Maddox, and P. Stone, "How humans teach agents," *Int. J. Social Robot.*, vol. 4, no. 4, pp. 409–421, 2012.

[60] G. Li, H. Hung, S. Whiteson, and W. B. Knox, "Learning from human reward benefits from socio-competitive feedback," in *Proc. 4th Joint IEEE Int. Conf. Develop. Learn. Epigenetic Robot.*, 2014, pp. 93–100.

[61] G. Li, S. Whiteson, W. B. Knox, and H. Hung, "Social interaction for efficient agent learning from human reward," *Auton. Agents Multi-Agent Syst.*, vol. 32, no. 1, pp. 1–25, 2018.

[62] T. Kaochar, R. T. Peralta, C. T. Morrison, I. R. Fasel, T. J. Walsh, and P. R. Cohen, "Towards understanding how humans teach robots," in *Proc. Int. Conf. User Model., Adaptation, Pers.*, 2011, pp. 347–352.

[63] R. T. Peralta, T. Kaochar, I. R. Fasel, C. T. Morrison, T. J. Walsh, and P. R. Cohen, "Challenges to decoding the intention behind natural instruction," in *Proc. IEEE Int. Symp. Robot Human Interact. Commun.*, 2011, pp. 113–118.

[64] G. Li, R. Gomez, K. Nakamura, and B. He, "Interactive reinforcement learning from demonstration and human evaluative feedback," in *Proc. 27th IEEE Int. Symp. Robot Human Interact. Commun.*, 2018, pp. 1156–1162.

[65] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *Proc. 21st Int. Conf. Mach. Learn.*, 2004, pp. 1–8.

[66] G. Li, H. Hung, S. Whiteson, and W. B. Knox, "Using informative behavior to increase engagement in the TAMER framework," in *Proc. 12th Int. Conf. Auton. Agents Multi-Agent Syst.*, 2013, pp. 909–916.

[67] P. L. Vail, *Emotion: The On/Off Switch for Learning*. Rosemont, NJ, USA: Modern Learning Press, 1994.

[68] J. Broekens, "Emotion and reinforcement: Affective facial expressions facilitate robot learning," in *Proc. Artif. Intell. Human Comput.*, 2007, pp. 113–132.

[69] G. Li, H. Hung, and S. Whiteson, "A large-scale study of agents learning from human reward," in *Proc. 14th Int. Conf. Auton. Agents Multiagent Syst.*, 2015, pp. 1771–1772.

[70] G. Li, H. Dibeklioglu, S. Whiteson, and H. Hung, "Towards learning from implicit human reward," in *Proc. 15th Int. Conf. Auton. Agents Multiagent Syst.*, 2016, pp. 1353–1354.

[71] V. Veeriah, P. M. Pilarski, and R. S. Sutton, "Face valuing: Training user interfaces with facial expressions and reinforcement learning," in *Proc. Interact. Mach. Learn. Workshop*, 2016, pp. 1–7.

[72] B. Schölkopf, "Artificial intelligence: Learning to see and act," *Nature*, vol. 518, no. 7540, pp. 486–487, 2015.

[73] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[74] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.

[75] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4299–4307.

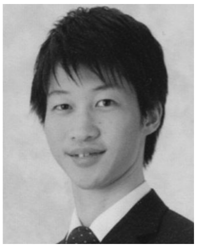
[76] M. K. Ho, J. MacGlashan, M. L. Littman, and F. Cushman, "Social is special: A normative framework for teaching with and learning from evaluative feedback," *Cognition*, vol. 167, pp. 91–106, 2017.

Guangliang Li (M'14) received the Bachelor's degree in automation and M.Sc. degree in control theory and control engineering from the School of Control Science and Engineering, Shandong University, Jinan, China, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from the University of Amsterdam, Amsterdam, The Netherlands, in 2016.

He was a Visiting Researcher with Delft University of Technology, Delft, The Netherlands, and a Research Intern with the Honda Research Institute Japan, Co., Ltd., Wako, Japan. He is currently a Lecturer with Ocean University of China, Qingdao, China. His research interests include reinforcement learning, human agent/robot interaction, and robotics.

Randy Gomez received the M.Eng.Sci. degree in electrical engineering from the University of New South Wales, Sydney, NSW, Australia, in 2002, and the Ph.D. degree in information science from the Graduate School of Information Science, Nara Institute of Science and Technology, Nara, Japan, in 2006.

He was a Researcher with Kyoto University, Kyoto, Japan, until 2012, under the auspices of the Japan Society for the Promotion of Science research fellowship. He is currently a Senior Scientist with the Honda Research Institute Japan, Wako, Japan. His research interests include robust speech recognition, acoustic modeling and adaptation, multimodal interaction, and robotics.



Keisuke Nakamura (M'09) received the B.E. degree in control and system engineering from the Department of Control and System Engineering, Tokyo Institute of Technology, Tokyo, Japan, in 2007, the M.E. degree in mechanical and control engineering from the Department of Mechanical and Control Engineering, Tokyo Institute of Technology, Tokyo, Japan, in 2010, and the Ph.D. degree in informatics from Kyoto University, Kyoto, Japan, in 2013.

He was with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, U.K., from 2007 to 2008. He is currently a Senior Scientist with the Honda Research Institute Japan, Co., Ltd, Wako, Japan. His research interests include robotics, control systems, and signal processing.

Bo He (M'18) received the Ph.D. degree in control theory and control engineering from Harbin Institute of Technology, Harbin, China, in 1999.

He was a Researcher with Nanyang Technological University, Singapore, from 2000 to 2002. He is currently a Full Professor with Ocean University of China, Qingdao, China. His research interests include SLAM, machine learning, and robotics.