The Augmented Intelligence Perspective on Human-in-the-Loop Reinforcement Learning: Review, Concept Designs, and Future Directions

Kok-Lim Alvin Yau, Yasir Saleem, Yung-Wey Chong, Xiumei Fan, Jer Min Eyu, and David Chieng

***Abstract*—Augmented intelligence (AuI) is a concept that combines human intelligence (HI) and artificial intelligence (AI) to leverage their respective strengths. While AI typically aims to replace humans, AuI integrates humans into machines, recognizing their irreplaceable role. Meanwhile, human-in-the-loop reinforcement learning (HITL-RL) is a semisupervised algorithm that integrates humans into the traditional reinforcement learning (RL) algorithm, enabling autonomous agents to gather inputs from both humans and environments, learn, and select optimal actions across various environments. Both AuI and HITL-RL are still in their infancy. Based on AuI, we propose and investigate three separate concept designs for HITL-RL: *HI-AI*, *AI-HI*, and *parallel-HI-and-AI* approaches, each differing in the order of HI and AI involvement in decision making. The literature on AuI and HITL-RL offers insights into integrating HI into existing concept designs. A preliminary study in an Atari game offers insights for future research directions. Simulation results show that human involvement maintains RL convergence and improves system stability, while achieving approximately similar average scores to traditional *Q*-learning in the game. Future research directions are proposed to encourage further investigation in this area.**

***Index Terms*—Artificial intelligence (AI), augmented intelligence (AuI), human in the loop, reinforcement learning (RL).**

I. Introduction

Augmented intelligence (AuI), or hybrid intelligence, is a concept that promotes human–machine collaboration [1], [2], enhancing performance compared to separate human and artificial intelligence (AI) approaches [3], [4]. Combining human intelligence (HI) and AI enables shared control and autonomy between humans and machines. Henceforth, *AI* and *machines* will be used interchangeably, as will *HI* and *humans*. Advantages include: 1) offloading cognitive burden from humans to machines in complex tasks [5]; 2) catering to personalized human needs and characteristics, such as personalized hearing aid settings [6] and manufacturing systems [7]; 3) minimizing errors by humans or machines [6]; and 4) improving human–machine coordination toward common goals [8]. Human involvement in AI is crucial for human-centered tasks and critical systems, where human inputs guide machines to improve machine actions and human capabilities, such as acquiring new skills in assistive technologies [9]. For instance, in [10], machines make local decisions followed by humans making final decisions, improving the accuracy of person identification and anomaly detection.

Through collaborative synergy, humans and machines leverage their strengths toward common goals, improving system performance. Humans and machines learn and make decisions differently: humans incorporate prior and new knowledge, considering subjective parameters such as intuition and experience for intuitive decision making, while machines start from scratch, relying on objective parameters such as measurable metrics [11]. The strengths of HI and AI are presented in Table I. Both HI and AI have shortcomings. First, while humans are generally considered experts in most AuI use cases, their knowledge levels vary, and they may not consistently provide accurate information. Second, both HI and AI can introduce biased decisions. Human bias may arise from behaviors like favoritism and racism, while machine bias can stem from environmental noise during the learning process. Using relative human inputs instead of absolute ones has been shown to mitigate biases [6]. This approach entails humans selecting a preferred option from multiple choices rather than providing open-ended responses. In addition, HI has two main disadvantages. First, there is a higher and inconsistent response delay from humans when providing inputs to machines. Second, human inputs are susceptible to noise [12].

AuI is a concept initially developed with a broad focus on AI, without specific emphasis on particular AI algorithms, including reinforcement learning (RL) [13] and deep RL (DRL) [14]. Meanwhile, many AI algorithms, including RL and DRL, were developed without considering human involvement. The integration of HI into RL and DRL algorithms has led

Received 22 August 2024; accepted 20 September 2024. Date of publication 18 October 2024; date of current version 20 November 2024. This work was supported by the Ministry of Higher Education, Malaysia, through the Fundamental Research Grant Scheme under Grant FRGS/1/2022/ICT02/UTAR/01/2. This article was recommended by Associate Editor B. Guo. *(Corresponding author: Yung-Wey Chong.)*

Kok-Lim Alvin Yau and Jer Min Eyu are with the Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman, Kajang 43200, Malaysia (e-mail: yaukl@utar.edu.my; jermin@1utar.my).

Yasir Saleem is with the Department of Computer Science, Aberystwyth University, SY23 3DB Aberystwyth, U.K. (e-mail: yasir.saleem@aber.ac.uk).

Yung-Wey Chong is with the School of Computer Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia (e-mail: chong@usm.my).

Xiumei Fan is with the School of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, China (e-mail: xmfan@xaut.edu.cn).

David Chieng is with the Department of Electrical and Electronic Engineering, Faculty of Science and Engineering, University of Nottingham Ningbo, Ningbo 315100, China (e-mail: David.Chieng@nottingham.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/THMS.2024.3467370.

Digital Object Identifier 10.1109/THMS.2024.3467370

TABLE I
STRENGTHS OF HI AND AI

HI	AI
• Higher capability in learning: • under a high-noise environment (e.g., limited/ inaccurate communication and sensing [15], [16]). • based on unquantifiable data (e.g., emotion) [17]. • based on arbitrary data (e.g., diverse unlabeled data) • soft skills (e.g., flexibility and imaginative). • independently and thinking out of the box. • Interpret complex patterns and images (e.g., diagnostic images [16], [18]). • Provide new goals (e.g., when the environment changes [10], [15]). • Enable transfer learning to transfer knowledge between situations [10].	• Higher computing capability for large data processing [10]. • Higher storage capacity [10]. • Longer involvement period than human (e.g., one hour [19]). • Provide real-time response. • Higher capability in learning complex, heterogeneous tasks [20]. • Higher consistency and efficiency.

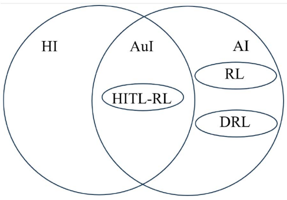


Fig. 1. Relationship between the AuI concept and the algorithms of RL, DRL, and HITL-RL.

to the human-in-the-loop reinforcement learning (HITL-RL) algorithm [5]. Fig. 1 illustrates the relationship between AuI, RL, DRL, and HITL-RL. While both the HITL-RL algorithm and the AuI concept involve human participation, their significance has not been extensively studied in the literature. This article aims to bridge this gap by reviewing HITL-RL from the AuI perspective, enabling the integration of AuI to enhance HITL-RL algorithms. Central to AuI are three separate concept designs dictating the sequence of HI and AI involvement: HI followed by AI (HI-AI), AI-HI, and parallel-HI-and-AI. Current HITL-RL algorithms, including reward shaping, policy shaping, guided exploration, and augmented value function (see Section I-C), typically align with AuI's parallel-HI-and-AI approach. In this article, we categorize HITL approaches based on AuI, aligning HITL-RL algorithms with AuI's three conceptual designs. This allows us to utilize AuI advancements, especially HI-AI and AI-HI approaches, to enhance HITL-RL further. After establishing an understanding of AuI, we provide an overview of RL, DRL, and HITL-RL and then outline the contributions and organization of this article.

A. Reinforcement Learning

Traditional RL enables an autonomous agent (or decision maker) to learn: 1) without being taught by an external teacher (*being unsupervised*); 2) without relying on mathematical models of humans and the environment (*being model-free*); and 3) during normal operation without prior training (*being online*) [13]. Compared to other algorithms, such as genetic algorithm, simulated annealing, and evolutionary programming, RL is easier to integrate with humans due to its unsupervised online nature [21]. RL has three main representations: state s_t for decision-making factors, action a_t, and reward $r_{t+1}(s_{t+1})$ for environment feedback. The agent observes state s_t and selects action a_t at time t (see Fig. 2). Subsequently, it receives reward $r_{t+1}(s_{t+1})$ and updates Q-value $Q_t(s_t, a_t)$, estimating future rewards for the state–action pair at time $t+1$. During action selection, the agent performs either *exploration* or *exploitation*. Exploration selects random actions at the exploration probability ε to learn Q-values for different actions in a state, so inappropriate actions may incur a long learning time and high negative rewards. Exploitation selects the best known action a_t^* with the maximum Q-value at probability $1-\varepsilon$ to maximize cumulative rewards over time. The exploration probability can decay from a higher initial value at time t_s to a lower value at t_e, specifically $\varepsilon = \varepsilon/(t_e - t_s)$.

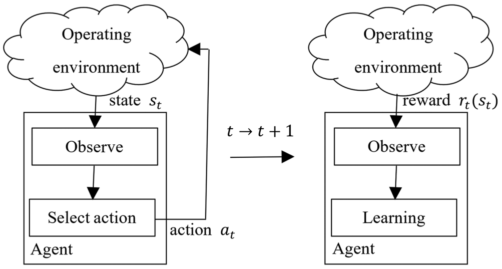


Fig. 2. RL agent performs action selection at time t and learning at time $t+1$. Decisions are made without human involvement.

Using the tuple $(s_t, a_t, r_{t+1}(s_{t+1}), s_{t+1})$, the agent updates Q-value $Q_t(s_t, a_t)$ using Q-function. The agent interacts with the environment and updates Q-values for all state–action pairs over time. Overall, the agent learns the optimal policy π_t^* to maximize cumulative rewards by selecting the right action for each state. HITL-RL integrates HI into traditional RL's learning or action selection phases. Section II-A presents RL variants, including Q-learning and SARSA, integrated with HI, highlighting their distinct Q-functions.

B. Deep Reinforcement Learning

DRL combines RL and deep learning (DL). The deep Q-network (DQN) is a popular DRL algorithm for estimating Q-values across various states and actions [14]. DRL utilizes neural networks with: 1) the input layer receives and transfers states; 2) hidden layers conduct nonlinear transformations using weights and activation functions, storing network parameters; and 3) the output layer estimates Q-values for all actions in a state [22]. Examples of layers are as follows: 1) the fully connected layer connects each neuron to all neurons in the next layer; and 2) the convolutional layer extracts spatial features from input data using filters. Compared to RL, the DQN offers three main advantages: 1) higher storage capacity; 2) using continuous state space that represents numerous states; and 3) higher convergence rate. Section II-B presents DRL variants integrated with HI, including DQN, policy gradient, model-based RL, inverse RL, and actor–critic RL.

C. Human-in-the-Loop Reinforcement Learning

The advantages and shortcomings of HITL-RL are based on the broader discussion in Section I. HITL approaches incorporate human inputs (e.g., feedback on human policy) for decision making during normal operation, without requiring a complete prior dataset [23], [24]. Humans may collaborate with agents physically through direct observation or remotely via video streaming [7]. Human inputs help agents to customize behavior and adapt to dynamic human needs, perceptions, and physical activities (e.g., walking patterns). Human inputs can be integrated into HITL-RL through four ways [25]: 1) *reward shaping* combines human rewards with the agent reward function [26], [27], [28]; 2) *policy shaping* integrates human actions into the agent's policy [29]; 3) *guided exploration* uses human inputs to guide the agent in exploration, increasing the convergence rate [30]; and 4) *augmented value function* updates the value function or Q-function using human inputs [31].

Human inputs offer three main advantages to HITL-RL. First, the need for a large number of training samples in RL and DRL can be lessened [11], [16], [32]. This reduces the effects of missing, noisy, unwanted, and irrelevant training samples, enhancing the quality of learned knowledge [16]. Second, HITL-RL consistently achieves a higher convergence rate compared to non-HITL approaches because human inputs reduce the search space for optimal actions [16], [33]. Third, learning relies on data collected from running policies in real environments [11]. As an example, in a gait-based person identification model [34], real-time adjustments are necessary as environmental changes directly impact gait, affecting identification accuracy. However, HI has two main disadvantages for HITL-RL. First, inaccurate human inputs reduce the convergence rate [35]. Second, human inputs are subjective and susceptible to undesirable behaviors (e.g., adversarial, uncertain, and biased).

D. Contributions of This Article

While both AuI and HITL-RL are emerging fields, this article contributes to the literature by examining HITL-RL from the AuI perspective, leveraging AuI approaches and concepts to enhance HITL-RL. There are four main contributions. First, three separate main concept designs (HI-AI, AI-HI, and parallel-HI-and-AI) for HITL-RL are presented based on the AuI concept. Second, state-of-the-art HITL-RL algorithms are reviewed based on these concept designs, providing a comprehensive view of human involvement in AI. Numerous related works have been published. HITL has been extensively reviewed from various perspectives of machine learning, including data processing, model training and inference, and applications, in [17]. Various integration methods include incorporating *human actions* [36] and different types of human inputs (e.g., facial expression, speech, and hand gestures) into reward functions [37]. An HITL DL framework in [38] provides *anomaly prediction and safe learning*. A comprehensive framework, covering humans, machines, tasks, and the kernel, in [10] allows humans to provide guidance (i.e., the change direction) for optimizing machine performance. Focusing on medical applications, an HITL-RL framework in [25] includes three main stages (data producing and preprocessing, model design, and evaluation) and state-of-the-art approaches [25], while Holzinger [16] reviews HITL applications. This article offers a novel perspective on HITL-RL by integrating human inputs based on the three concept designs. Third, the performance of the three concept designs, including the convergence properties and system performance metrics (the instantaneous and standard deviation of game scores), is investigated. Fourth, future research directions for improving the three concept designs are presented to simulate interest in this emerging topic.

E. Organization of This Article

The rest of this article is organized as follows. Section II presents background revolving around HI and D(RL). Section III presents three concept designs based on AuI. Section IV presents categories of parallel-HI-and-AI HITL-RL. Section V presents simulation results comparing the concept designs. Section VI presents future research directions. Finally, Section VII concludes this article.

II. Background of Integrating HI Into (D)RL

A. Variants of RL Algorithms

1) Q-Learning: The Q-function relies on the maximum Q-value of the next state s_{t+1} under the assumption that the agent selects the optimal action in any state [13]

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \Big[r_{t+1}(s_{t+1}) + \gamma \max_{a \in A} Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \Big] \tag{1}$$

where $0 \leq \alpha \leq 1$ is the learning rate and $0 \leq \gamma \leq 1$ is the discount factor.

2) SARSA: The Q-function relies on the Q-value $Q_t(s_{t+1}, a_{t+1})$ of the next state s_{t+1} after executing both the current action a_t and the next action a_{t+1}, selecting a more conservative (safer) action compared to Q-learning [13]

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha \left[r_{t+1}(s_{t+1}) + \gamma Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t) \right]. \tag{2}$$

B. Variants of DRL Algorithms

1) Deep Q-Network: The DQN comprises two deep neural networks: the main network θ_t and the target network θ_t^-, which is a clone of the main network. The agent observes state s_t and inputs it into the main network to estimate Q-values $Q_t(s_t, a_t; \theta_t)$ for all potential actions $a_t \in A$. It selects the optimal action a_t^* with the highest Q-value $Q_t(s_t, a_t; \theta_t)$ at time t (see Fig. 3). The agent receives a reward $r_{t+1}(s_{t+1})$ from the environment at the time $t + 1$. This transition $(s_t, a_t, r_{t+1}(s_{t+1}), s_{t+1})$ is stored as experience in the *replay memory*, which holds a large number of experiences (e.g., 200 000 [22]). During training, the agent uses a minibatch of N_e experiences e_i and the target network to generate target Q-values $Q_t(s_t, a_t; \theta_t^-)$ used in estimating the target $y_i = r_{i+1}(s_{i+1}) + \gamma \max_a Q_i(s_{i+1}, a; \theta_i^-)$ for calculating the loss function $L(\theta_i) = E_{s,a}[(y_i - Q_i(s, a; \theta_i))^2]$ used

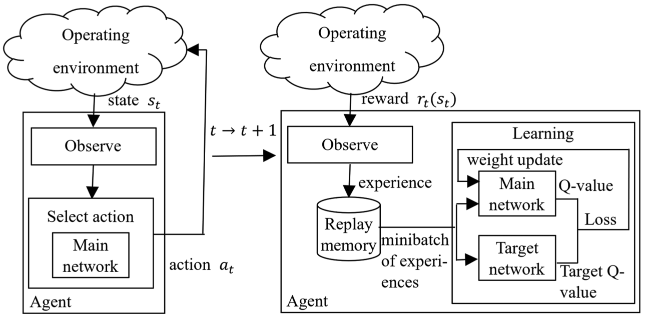


Fig. 3. DQN agent selects an action using the main network at time t and learns using main and target networks along with replay memory at time $t+1$. Decisions are made autonomously, without human involvement.

in updating the main network parameters $theta_i$ through backpropagation and stochastic gradient descent $\nabla_\theta L(\theta_i)$. The target network is duplicated from the main network every C iterations.

2) Policy Gradient: Policy gradient parameterizes policy $\pi(a|s_t;\theta_t)$ and directly optimizes the policy parameters θ_t without using a Q-function. Using the policy $\pi(a|s_t;\theta_t)$, the agent generates a probability distribution over the action space, assigning a probability to each potential action $a \in A$ given a state s_t. The probabilities sum to 1, and actions with higher probabilities are more likely to be selected. The agent receives a reward $r_{t+1}(s_{t+1})$ from the environment at time $t+1$. The agent stores trajectories $\tau_1, \tau_2, \ldots, \tau_n$, where each trajectory $\tau = \{(s_0, a_0, r_1(s_1)), (s_1, a_1, r_2(s_2)), \ldots, (s_{\tau-1}, a_{\tau-1}, r_\tau(s_\tau))\}$ is a sequence of states, actions, and rewards in an episode. During training, given the trajectories, the agent computes the gradient $\nabla_\theta J(\theta) = E_{\tau\sim\pi(a|s_t;\theta_t)}[\sum_{t=0}^{T} \nabla_\theta \log \pi(a|s_t;\theta_t) A_t]$ and uses stochastic gradient ascent to update the policy parameters θ_t to maximize the expected return. The advantage function A_t represents the difference between the return of a state–action pair and the average return of all actions given a state s_t. Proximal policy optimization (PPO) extends policy gradient to limit the extent of policy updates to improve stability.

3) Model-Based RL: Model-based RL learns human and environment models, which capture their dynamics, used to estimate rewards and next states, and select actions through internal simulations. Details on human and environmental models are presented in Section II-E.

4) Inverse Reinforcement Learning (IRL): IRL learns a reward function $r(s,a)$ from expert demonstrations D, which are sequences of states and actions, assigning higher rewards to actions likely selected by the expert policy $\pi_{\text{expert}}(a|s)$. Using maximum entropy IRL, the agent learns the reward function $r(s,a)$ to maximize the objective function $\max_r \sum_{(s,a)\in D} \log(\pi_{\text{expert}}(a|s)) - \lambda \sum_s \sum_a \pi_{\text{agent}}(a|s) \log(\pi_{\text{agent}}(a|s))$, which maximizes the log-likelihood of expert demonstrations and the entropy of the agent's policy through penalizing deterministic actions. This aligns its policy $\pi_{\text{agent}}(a|s)$ with the expert policy $\pi_{\text{expert}}(a|s)$, while maximizing the entropy of the policy $\pi_{\text{agent}}(a|s)$ for increased randomness (or diversity). Subsequently, the agent uses the learned reward function $r(s,a)$ to train policy $\pi_{\text{agent}}(a|s)$ using traditional DRL algorithms [39].

5) Actor–Critic RL: Actor–critic separates Q-value update and temporal difference calculation to increase the convergence rate in traditional RL. Two components that interact with each other are as follows: 1) the actor θ^π selects actions and learns the optimal policy to maximize Q-values over time, and 2) the critic θ^Q evaluates the quality of the actor's selected action. The soft actor–critic algorithm extends traditional actor–critic with entropy regularization for increased randomness and stability, leading to an even higher convergence rate. The actor computes the gradient $\nabla_{\theta^\pi} J(\theta^\pi, \theta^Q) = E_{s_t\sim D}[\nabla_{\theta^\pi} \log \pi(a_t|s_t;\theta_t) \cdot \nabla_{a_t} Q(s_t, a_t; \theta^Q)]$ and updates the policy parameters θ^π through stochastic gradient ascent to maximize cumulative reward. The critic computes the gradient $\nabla_{\theta^Q} L(\theta^Q) = \nabla_{\theta^Q} E_{(s_t,a_t,r(s_t,a_t),s_{t+1})\sim D}[(Q(s_t,a_t;\theta^Q) - y_t)^2]$ and updates the policy parameters θ^Q through stochastic gradient ascent to minimize the mean squared Bellman error $L(\theta^Q)$.

C. Sources and Main Roles of HI in (D)RL

Human inputs are integrated into state, action, and reward representations in HITL-RL [40], such as: 1) states incorporated with hand and knee movements [43], [44], human images [45], and human verification outcomes [40]; 2) actions provided through devices such as keyboard and mouse [26], and hand gestures such as hand and body movements [46]; and 3) rewards incorporated with facial and emotional expressions [41], [42] and instructions and advice in natural languages [41], [47]. For instance, in the human–robot interaction application [41], [42], human inputs (e.g., facial and emotional expressions) are mapped to varying reward levels, updating Q-values accordingly.

Human inputs are crucial in training and action selection. During training, humans: 1) monitor system performance, and identify errors and inappropriate actions to prevent them in the future [19]; and 2) learn human-centered objective functions, which are subjective and difficult to be digitized [5]. During action selection, humans guide agents in: 1) searching for optimal actions at the early phase to increase the learning speed [5] and 2) handling and completing complex tasks under unstructured environment [5], [48].

D. Human Rewards and Environmental Rewards in (D)RL

In traditional RL, environmental rewards are utilized, whereas in HITL approaches, human rewards are incorporated to reflect human feedback and preferences. Environmental rewards are objective and measurable such as temperature [22], whereas human rewards are subjective and measurable/nonmeasurable such as human perception [22]. Both human and environmental rewards contribute to achieving the shared goal of humans and agents. The environmental reward is crucial in the absence of human rewards or when the environment is unfamiliar to humans.

E. Human Models and Environment Models in (D)RL

While human involvement in training and action selection leverages the strengths of HI (refer to Table I), continuous interaction with a real human can be labor intensive and challenging [6], [49], [50]. Human involvement typically spans short periods, for instance, 1 h [19] or 4.5 h [51], encompassing various tasks such as gathering human inputs and training the agent.

Hence, digital twins—models of humans and the environment—aim to approximate the true behaviors and dynamics of both [22], subsequently utilized in training and action selection through internal simulations. The human model predicts human states [52] and rewards [22] by mimicking human interactions, while the environment model predicts environment states and rewards by simulating environmental dynamics. The human model can be acquired by observing real human actions and responses in the actual environment.

There are three examples of human models. First is the *numerical reward-based human model*. In [53], the model consists of state–action pairs, with humans assigning a numerical human reward to each pair based on the long-term consequences of actions in given states. Second is the *action probability-based human model*. In [22], the model consists of state–action pairs, with humans assigning a probability of selection to each pair based on observations of human actions over a long term (i.e., four weeks). Third is *historical behavior-based human model*. In [52], the model is built using long short-term memory (LSTM) [54]. LSTM, a recurrent neural network, learns from historical data, capturing long-term dependencies in data sequences by managing information in its memory cells. LSTM predicts the next state based on environment dynamics [52] during both training and action selection.

There are two examples of environment models. First is *historical data-based environment model*. In [22], the model characterizes indoor temperature dynamics using Newton's law of cooling and heating, considering the temperatures from the fan coil and outdoor environment, along with historical temperature trends for a given day. Second is the *human preference-based environment model*. In [6], the model consists of numerous (o_1, o_2, μ) sets (e.g., 200), with humans assigning human preference μ to each set representing whether option o_1 or o_2, or both, is preferred for the sets of options (o_1, o_2). For generalization, the agent swaps features among a batch of (o_1, o_2, μ) sets. The model combines a convolutional neural network (CNN) with a bidirectional LSTM to minimize the loss between actual human inputs and predicted rewards.

The agent interacts with either the real human and environment or digital twins. In real environments, an agent executes one action per iteration, while in digital twins, the agent can execute multiple actions per iteration (refer to Section II-F1), increasing learning and reducing the need for real-world interactions between human and environment [37]. However, creating these models presents four challenges [11]: 1) acquiring a large number of training samples from humans and the environment [52]; 2) uncertain prior knowledge about the required training samples quantity; 3) potential inadequacy of training samples in representing the full spectrum of environmental dynamics [52]; and 4) long time is needed for model creation.

F. Key Considerations of Human Involvement in HITL-RL

1) Synchronous and Asynchronous Learning Approaches: In the synchronous approach, an agent updates its knowledge immediately after taking an action in the real environment, whereas in the asynchronous approach, the agent updates its knowledge separately in between interactions with the real environment, using human and environment models (see Section II-E) [11]. Consequently, the asynchronous approach involves more update steps, resulting in quicker learning and reduced human involvement.

2) Estimating the Optimal Policy for a New Task: An agent can generalize its learned knowledge across multiple tasks, enabling it to derive the optimal policy for a new task, thereby minimizing human involvement. This expands upon traditional D(RL), enabling the agent to learn a dedicated policy for every single task. In [49] and [50], action selection relies on three factors: the current state, the task policy, and task parameters containing specific task details. For generalization, given a particular current state, the agent learns the optimal generalized task policy to minimize the long-term average cost across various tasks with distinct parameters. Given a new task, the agent estimates the optimal policy and selects actions accordingly. Through gradient-based nonconvex optimization, RL enhances its ability to estimate the optimal policy for new tasks based on learned knowledge.

3) Considering Human Experience and Types of Human Inputs When Interacting With Agents: Human inputs may not always be beneficial in HITL-RL. When humans interact with agents, their experience influences their perception of the agents and their willingness to continue future interactions. In [55], humans offer two types of inputs for agents: critiques (e.g., "good job" or "don't do that") and actions (e.g., moving left or right). Agents learn from these inputs whether to repeat similar actions in the future. However, diverse critiques can lead to confusion between a critic and the expected action. Hence, the agent can replace its selected actions with human actions.

There are three distinctions in human experience between using human critiques and actions. First, human actions enable agents to select the optimal agent action in the *next* time instant, whereas human critiques assess agent actions from the *past* without suggesting future actions. Second, using human actions to select the optimal agent action in the next time instant incurs a response delay, whereas this delay is inconsequential with human critiques focused on past actions. Third, applying immediate positive or negative rewards to the right past actions with human critiques is challenging due to multiple past actions. However, the agent's action at the next time instant is unaffected by human critiques even if they are: 1) not received; 2) misinterpreted; or 3) lacking clarity on the right action(s) affected by a specific human critique. Utilizing human actions has demonstrated greater human friendliness. Experimental findings indicate that using human actions enhances human experience and average reward, while reducing training time and the number of time steps to complete a task.

G. Applications of HITL-RL

In HITL-RL, human inputs are incorporated to perform tasks and solve problems. Table II presents HITL-RL applications, covering the AuI approaches, the functions and human roles, HITL-RL models, and the performance improvement from HITL-RL. The table provides insights into how HITL-RL can be used in various applications. Similar to traditional RL and DRL, HITL-RL models have three main representations: 1)

TABLE II
EXAMPLES OF THE APPLICATION OF AuI-BASED HITL-RL ALGORITHMS

Application	Function	Human input	Human input type			Agent Model			Performance
			State	Action	Reward	State	Action	Reward	
Thermal management in building [22]	DQN adjusts room temperature based on human inputs and environmental rewards (difference to desired temperatures).	Temperature adjustments			✓	Environment (temperature and humidity), occupancy, activities, and energy consumption.	Temperature adjustment.	A weighted sum of human and environmental rewards.	Reduced human adjustments and deviation from desired indoor temperatures.
Target reaching [19]	Q-learning guides a robot to locate and reach a human-preferred target.	EEG signals			✓	Robot's position.	Robot movement (up, down, left, right).	Reward reduces with undesired human EEG signals.	Higher convergence rate and accuracy.
Weight lifting [49], [50]	Model-based RL guides an exoskeleton robot to assist a human in lifting weights.	EMG signals		✓		Robot's joint angle and velocity.	Joint torques, air pressure, and human's muscle activation.	Reward increases as EMG decreases.	Reduced human effort in weight lifting.
Human-robot interactions [56]	DQN controls a robot to recognize and respond to human emotions, gestures, and speech.	Accept/reject robot responses		✓	✓	Human facial expression, hand gesture, and speech.	Robot's speech, facial, and gestural outputs via LED display.	Human feedback on robot's action.	Higher convergence rate and shorter response time.
Catastrophe avoidance [51]	DQN guides an agent to avoid inappropriate actions that cause poor performance (catastrophe).	Correct agent actions and assign rewards		✓	✓	Agent's paddle position.	Paddle movement.	A weighted sum of human penalties and environmental rewards.	Higher average reward.
Vehicle energy management [48]	Q-learning guides simulated vehicles to achieve desired speed with minimal energy consumption.	Scores indicating the appropriateness of actions			✓	Vehicle's battery level, speed, and acceleration.	Power and power distribution of the internal combustion engine.	Reward increases with energy efficiency and human scores.	Higher average reward and reduced energy consumption.
Ball balancing [57]	PPO guides a robot to assist humans in balancing a ball on a board, learning from human decisions.	Balance a ball on a board			✓	Ball's coordinates.	Board tilting.	Reward increases as robot visits high-reward states and mimics human actions.	Reduced human effort in balancing the ball.
Object detection [58]	DQN guides a drone to gather valuable training data for object detection and tracking.	Select reward values		✓	✓	Drone and object coordinates.	Drone movement and human action requests.	Reward increases as human input decreases.	Higher precision and reduced human effort.
Virtual assistant [59]	Policy gradient responds to human requests (e.g., musics) based on human suggestions.	Select reward values			✓	Human requests.	Human suggestions (a music).	Human satisfaction, reward becomes negative when human stops the music.	Higher accuracy in classifying requests.
Personalized hearing aids [6]	CNN with Q-learning guides an agent in audio compression based on hearing aid preferences.	Use environment model to predict rewards			✓	Noisy audio.	Audio compression with different frequency bands.	Human satisfaction with the compressed audio.	Higher average reward (human preference).
Gesture imitation [32]	PPO guides a robot to fine-tune the trained reward function.	Correct the reward function			✓	Human gestures.	The robot movement.	Reward increases with gesture similarity to humans.	Higher average reward (more natural movement).
Mobility assistant [52]	Model-based RL guides a robot to predict human needs in motion support.	Motion intention and gait		✓		The positions and velocity of human and robot.	The robot movements.	Reward increases as movement cost decreases.	Higher average reward and prediction accuracy.

TABLE II
(CONTINUED)

Application	Function	Human input	Human input type			Agent Model			Performance
			State	Action	Reward	State	Action	Reward	
Obstacle avoidance [33], [35], [60]	Q-learning, SARSA, soft actor–critic, and PPO guide a robot navigate obstacles with human feedback, respectively.	Control the robot movement		✓		Robot's position, velocity, distance to obstacles, and obstacle positions.	The robot movement.	Reward increases on reaching the destination and decreases on hitting obstacles.	Higher average reward and convergence rate.
Object interaction [11], [61]	Model-based RL [61] and CNN with policy updates [11] guide a robot to open and close objects [11], [61] using human images, demonstrations [61], and feedback [11].	Provide relevant states for exploration and label datasets with rewards	✓		✓	Robot and door positions.	The robot's joint and gripper movements (left, right, open, close).	Reward increases as the exploration rate increases [61] and human satisfaction (reduced human corrections) [11].	Higher average reward [61] and success rate [11].
Car parking [39]	IRL guides a car to park using human demonstrations when it exceeds step limits.	Provide demonstrations	✓			Car position and orientation, and human demonstrations.	The car movement (left, right, up, down, rotate).	Reward increases with action similarity to humans.	Reduced human effort in demonstration.
Person identification [34]	DQN enables a base tree-structured model to classify humans as genuine or impostor.	Feedback on identification accuracy	✓			Model structure and human feedback.	Model (node probabilities) update.	Reward is positive (negative) for (in)correct identification.	Higher accuracy in identifying impostor.

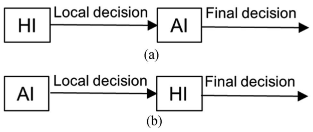


Fig. 4. AuI approaches. (a) HI-AI. (b) AI-HI.

state s_t representing decision-making factors; 2) action a_t, such as adjusting dynamic weight factors for balancing variables in an objective function [48] and control parameters [62]; and 3) reward $r_{t+1}(s_{t+1})$ representing system performance. Modeling the reward using system performance helps the agent improve the overall system rather than focusing on individual factors.

TABLE III
ADVANTAGES OF HI-AI AND AI-HI HITL-RL ALGORITHMS

HI-AI	AI-HI
Leverage human strengths (e.g., out-of-the-box thinking) in initial decision-making (see Table I).	Leverage AI strengths (e.g., high storage capacity) in initial decision making (see Table I).
HI offers context-aware inputs leveraging human insights in initial decision-making.	AI explores diverse possibilities using extensive data in initial decision making.
Agent corrects human biases/ errors before final decisions.	Human corrects agent biases/ errors before final decisions.

III. CONCEPT DESIGNS OF HITL-RL: THE AUI PERSPECTIVE

Based on the order in which decisions are made in the AuI concept, there are three separate approaches: 1) *HI-AI* where humans make local decisions and then machines make final decisions [see Fig. 4(a)]; 2) *AI-HI* where machines make local decisions and then humans make final decisions [see Fig. 4(b)]; and 3) *parallel-HI-and-AI* where humans and machines make final decisions simultaneously [see Fig. 5(c)]. HI-AI and AI-HI are cascading approaches, with AI more reliable in HI-AI and HI more reliable in AI-HI. In parallel-HI-and-AI, both HI and AI are equally reliable. In all three approaches, humans and machines interact with each other in a loop. The rest of this section presents concept designs for (D)RL, structured according to the purposes of these designs. Table III compares the advantages of HI-AI and AI-HI approaches. Table IV summarizes AuI-based HITL-RL algorithms, including their purposes, functions, features, and performance. The selection of HITL-RL algorithms for specific applications can be guided by the functions and features presented in the table.

A. *HI-AI-Based HITL-RL*

HI-AI, a traditional AuI approach, has been applied with the genetic algorithm [63], CNN [64], [65], and RL [66], [67]. For instance, in brain–computer interfaces [64], humans provide physiological signals, and the CNN generates commands representing human intentions. In manufacturing [63], humans offer inspiration, and genetic algorithms provide suggestions for new product designs. In HI-AI HITL-RL, HI initiates by gathering observations from the environment and selecting human action $a_{h,t}$, as shown in Fig. 5(a). This action, combined with the agent state $s_{m,t}$, forms the overall state $s_t = (s_{m,t}, a_{h,t})$ used by the agent during training and action selection. Human decisions

TABLE IV
Summary of AuI-Based HITL-RL Algorithms

AuI	Purpose and Reference	Function/ feature	Performance
HI-AI	Receive human actions for decision making [49], [50]	Agent incorporates human actions in states and learns the optimal policy for minimizing the long-term cost.	Higher system performance.
AI-HI	Provide recommended agent actions for humans to select [56]	Agent provides recommended options and learns using human rewards. Human selects one of the options and provides human rewards.	Higher convergence rate.
	Verify agent actions with human inputs [51]	Humans correct agent actions and assign negative rewards to inappropriate actions, which the agent then incorporates.	Higher average reward without severe system performance.
Parallel (reward shaping)	Use a weight to combine human and environmental rewards [22], [48]	Use a weight to combine human and agent rewards using Eq. (3).	Higher system performance and performance stability.
	Use human inputs to select a reward function [19]	Use (un)desirable human inputs to select the right reward function.	Higher system performance.
	Use human inputs to determine a reward function [57]	Revise reward function based on human goal's posterior probability and features (i.e., exploring popular states with higher information density, matching human pace in exploration, and avoiding inappropriate states and actions).	Higher system performance and similarity between human and agent actions, and reduced human effort.
	Use human inputs to select reward values [58]	Use reward $r_{h,t+1}(s_{h,t+1}) = -1$ for each request for human actions to reduce the cumulative reward with increasing human involvement.	Fewer requests for human actions with an acceptable system performance degradation.
	Use human inputs to select reward values [59]	Use reward $r_{h,t+1}(s_{h,t+1}) = 1$ for appropriate agent action with human satisfaction.	Higher system performance.
	Use human inputs to replace the reward function [6]	Gather human preferences to create an environment model for predicting rewards during normal operation.	Higher system performance.
	Use human inputs to select reward values [32]	Calculate a reward value to capture differences between human and agent actions for learning minor details of complex tasks without using a perfect reward function.	Higher system performance.
Parallel (policy shaping)	Integrate human action and agent action [52]	Select a joint human−agent action so that the agent action is closer to the desired state.	Higher cumulative reward (or lower cumulative cost).
	Request for human action at uncertain and unfavorable circumstances [33], [60]	Detect outliers in states (e.g., events), actions (e.g., unreliable actions), and rewards (e.g., low reward values), and then request for human actions.	Higher cumulative reward.
	Use human inputs to select a subset of possible actions [35]	Use human perceptions to estimate action appropriateness, then select appropriate actions from the action space.	Higher cumulative reward and task execution success rate.
Parallel (guided exploration)	Use minimal human information to guide state exploration [61]	Use a small number of relevant states given by humans to learn the exploration policy.	Higher system performance and relevant state interactions.
Parallel (augmented value function)	Use human inputs to adjust loss function [11]	Replace the agent reward with human feedback to identify and rectify states and actions.	Higher task execution success rate.
	Learn reward values through demonstration [39]	Learn sub tasks of a complex task, and incorporate failure experiences into the reward function.	Higher system performance and reduced human effort.

enhance AI's context awareness of the environment, especially subjective parameters, leading to improved correctness of the final decision $a_{m,t}$ made by the agent.

1) Receiving Human Actions for Decision Making: The agent integrates human actions into its states. In [49] and [50], the overall state comprises the agent's state and human action. The agent learns the optimal policy to minimize the long-term cost (or negative reward) $r_{t+1}(s_{t+1})$ over time. The next state s_{t+1} is updated with the current state, agent action, and additive Gaussian noise. There are four main steps in training and action selection. First, the agent learns a human–environment model comprised of transition probabilities between states, each containing the agent state and human action. Second, the agent determines preliminary policies for the states. Third, the agent estimates the long-term cost of a policy based on this model. Fourth, the agent identifies the optimal policy with the least cost.

In a person identification model [34], the overall state includes the agent state and human action. The agent state comprises the current person identification model structure (i.e., characteristics and variables) and human feedback. The model is a base tree-structured classification model with trees having M leaf nodes providing negative probabilities $y_i \in (y_1, y_2, \ldots, y_M)$ of features i representing the person to be identified. An average negative probability $y = \frac{1}{M}\sum_{i=0}^{M} y_i$ is calculated, and a threshold determines a positive (negative) identification outcome indicating a genuine user (impostor). Utilizing the negative probabilities of the nodes, the model calculates the reliability of the identification outcome $\frac{1}{M}\sum_{i=0}^{M} |y_i - \frac{n_i}{p_i+n_i}|$, which is an average discrepancy between the negative probabilities and the ratio of the number of negative feedback n_i to all feedback $p_i + n_i$. Then, a threshold is used to determine whether a human expert should be requested to provide feedback. Human feedback indicates the correctness of identification outcomes, with $f = 1$ ($f = 0$) indicating correct (incorrect) identification.

B. AI-HI-Based HITL-RL

AI-HI, a traditional AuI approach, has been applied with the greedy algorithm [68] and random forest [69]. For instance, in the packaging process [68], the AI-based greedy algorithm increases the number of objects in a container, which humans

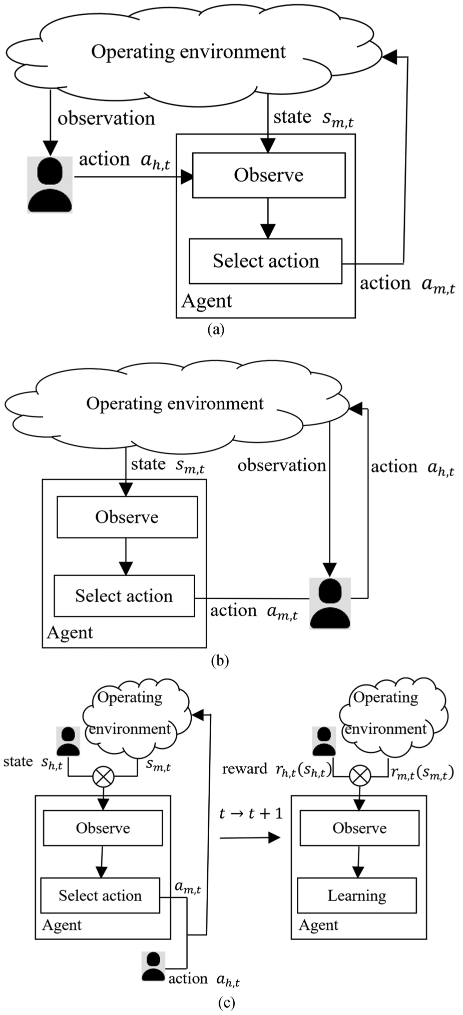


Fig. 5. AuI-based HITL-RL algorithms. (a) HI-AI HITL-RL. (b) AI-HI HITL-RL. (c) Parallel-HI-and-AI HITL-RL.

then refine to further optimize. Similarly, in fault detection [69], random forest detects faults based on sensor data, which humans then refine to further optimize. In AI-HI HITL-RL, AI initiates by observing agent state $s_{m,t}$ from the environment and selecting agent action $a_{m,t}$, as shown in Fig. 5(b). Humans receive the agent actions $a_{m,t}$, collect observations from the environment, verify and evaluate the quality of the agent actions based on subjective parameters, and select either the same action $a_{h,t} = a_{m,t}$ or a different action $a_{h,t} \neq a_{m,t}$ [70].

1) Providing Recommended Agent Actions for Humans to Select Human Actions: The agent recommends agent actions for humans to select from, reducing human effort in identifying the optimal option from many possibilities. In [56], the agent recommends four agent actions with the highest Q-values for humans to select from, or they can select a different action. Selecting a recommended agent action indicates the correct agent's selection. Human rewards replace environmental rewards: positive if a recommended action is chosen, and negative if none are chosen.

2) Verifying Agent Actions With Human Inputs: Humans verify agent exploration actions to prevent significant negative consequences [7], [51]. In [51], the agent stores state–action pairs with binary labels. During training, humans label inappropriate state–action pairs and replace these actions with appropriate ones. During action selection, humans avoid state–action pairs with inappropriate labels and use reward shaping (see Section IV-A) to assign significant negative rewards to them. The agent then learns to replace inappropriate actions autonomously without human involvement. This combined labeling and reward shaping has been shown to achieve higher average rewards than labeling alone.

C. Parallel-HI-and-AI-Based HITL-RL

In the parallel-HI-and-AI approach, humans and agents collaborate to provide information, learn from each other, and make decisions [see Fig. 5(a)]. Human entry points include state $s_t = (s_{h,t}, s_{m,t})$, action $a_t = f(a_{h,t}, a_{m,t})$, and reward $r_t(s_t) = f(r_{h,t}(s_{h,t}), r_{m,t}(s_{m,t}))$. Most traditional HITL-RL algorithms, including reward shaping, policy shaping, guided exploration, and augmented value function (see Section I-C), are based on parallel-HI-and-AI.

The state $s_t = (s_{h,t}, s_{m,t})$ combines human $s_{h,t}$ and the agent $s_{m,t}$ states to enhance the agent's context awareness. Human states can include measurements and sensing outcomes (e.g., brain signals or EEG), facial expression [56], heart rate, speech or sound [56], [71], body temperature, gestures [21], [56], and kinematics measurements [8], [62], [72], [73]. Real-time crowdsourced data from multiple humans provide distributed HI, and unstructured human inputs (e.g., feedback) are preprocessed into structured human states [74]. The action $a_t = f(a_{h,t}, a_{m,t})$ combines human $a_{h,t}$ and agent $a_{m,t}$ actions, both considered optimal from their perspectives. The actions and policies (e.g., two independent probabilities) can be combined through addition $a_t = a_{h,t} + a_{m,t}$ [21] or multiplication $\pi_t \propto \pi_{h,t} \times \pi_{m,t}$ [29]. Human actions $a_{h,t}$ may be requested irregularly and can be selected within the agent's action space like any other potential actions [58]. The reward $r_{t+1}(s_{t+1}) = f(r_{h,t+1}(s_{h,t+1}), r_{m,t+1}(s_{m,t+1}))$ combines human $r_{h,t+1}(s_{h,t+1})$ and the agent $r_{m,t+1}(s_{m,t+1})$ rewards [37], [75], [76], [77]. Nonmeasurable human rewards (e.g., discomfort [22]) can be estimated using measurable metrics like the frequency of humans adjusting the room temperature and the adjusted temperature needed for comfort.

IV. Categories of Parallel-HI-and-AI-Based HITL-RL Algorithms

This section presents the four HITL-RL algorithms.

A. Reward Shaping

Reward shaping integrates human rewards into the reward function [25], [27].

1) Using a Weight to Combine Human and Environmental Rewards: The reward function uses a weight w to adjust the contributions of human and agent rewards based on the confidence (or uncertainty) in actions selected by both in a given state as follows [9], [22], [48], [56]:

$$r_{t+1}(s_{t+1}) = w \cdot r_{m,t+1}(s_{m,t+1}) + (1 - w) \cdot r_{h,t+1}(s_{h,t+1}). \tag{3}$$

The proposed solution has been shown to improve system performance and stability as humans can be inconsistent and insensitive to the environment [22].

2) Using Human Inputs to Select a Reward Function: The right reward function can be selected from a set of potentials based on human inputs, which can be desirable (true) or undesirable (false). In [19], when the human provides a false signal, the reward function is $r_{t+1}(s_{t+1}) = r_{t+1}(s_{t+1}) \times i + j$, where $i < 1$ and $j < 0$ reduce the reward value. Otherwise, the constant reward function is $r_{t+1}(s_{t+1}) = k$, where $k > 0$ increases the reward value [19]. The proposed solution has been shown to improve system performance.

3) Using Human Inputs to Determine a Reward Function: The reward function is dynamic as human goals change over time, often becoming more specific. In [57], the agent integrates two main steps into DRL to update the reward function. First, given the current state $s_t = (s_{1,t}, s_{2,t}, \ldots, s_{s,t}, \ldots, s_{|s|,t})$, the agent estimates the posterior probability of the human goal g_s for each state component $s_{s,t}$ using state-based multivariate Bayesian inference [78], [79] that incorporates multiple goals $g_1, g_2, \ldots, g_s, \ldots, g_{|s|}$ and their respective prior distributions $P(g_1), P(g_2), \ldots, P(g_{|s|})$ to compute the posterior distribution $P(g_1, g_2, \ldots, g_{|s|}|s_t)$, which is the estimation of the joint distribution of all potential goals given observed states. Second, the agent revises the reward function $r_{t+1}(s_{t+1}) \propto \sum_{i=1}^{I} w_i \prod_{s=1}^{|s_t|} H_i(s_{s,t})$, which is the sum of the posterior probabilities of the human goals. $H_i(s_{s,t})$ represents a human feature, encouraging the agent to: 1) explore high-information density and frequently visited states; 2) follow the human pace in exploration; and 3) avoid inappropriate states and actions. The proposed solution enables agents to quickly adapt to the latest human goals, reducing the gap between agent and human goals, increasing similarity between human and agent actions, enhancing system performance, and reducing human effort.

4) Using Human Inputs to Select Reward Values: The reward value can be based on human inputs. In [58], human and agent rewards are combined using weights (see Section IV-A1). The agent receives a human reward $r_{h,t+1}(s_{h,t+1}) = -1$ when it requests human actions. This algorithm reduces the number of human action requests with acceptable system performance degradation. In [59], the agent receives a human reward $r_{h,t+1}(s_{h,t+1}) = 1$ for an appropriate action that brings human satisfaction and $r_{h,t+1}(s_{h,t+1}) = -1$ otherwise. The proposed solution has been shown to achieve a higher system performance.

5) Using Human Inputs to Replace the Reward Function: Designing an accurate reward function reflecting the real system performance can be challenging in complex tasks [6]. In [6], the agent collects human preferences to create an environment model (see Section II-E) and then receives predicted rewards from this model, rather than the real environment. In [32], the agent learns minor details overlooked by the reward function. Using DRL, the robot agent learns to imitate human gestures (e.g., saluting and waving). The reward function calculates the cosine similarity between the human and agent action vectors $r_{t+1}(s_{t+1}) \propto \cos(\frac{a_{h,t} \cdot a_{m,t}}{\|a_{h,t}\|\|a_{m,t}\|})$, which is the cosine of the angle between the human action vector $a_{h,t}$ and agent action vector $a_{m,t}$. Given its inherent inaccuracy, the reward function is imperfectly accurate. So, the agent interacts with real human actions to fine-tune the agent policy learning latent vectors to further minimize discrepancies between human and agent actions, thereby generating more human-like motions [32]. The reward function measures the difference between the human and agent actions $r_{t+1}(s_{t+1}) \propto \|a_{h,t} - a_{m,t}\|_2$, incorporating latent vectors. The proposed solution has been shown to reduce training samples with human involvement and enhance system performance. A similar algorithm is found in [80].

B. Policy Shaping

Policy shaping integrates human actions, considered optimal from the human perspective, into the agent policy.

1) Integrating Human Action and Agent Action: The action a_t incorporates human action $a_{h,t}$ and agent action $a_{m,t}$. In [52], the agent uses a human model (see Section II-E) to estimate the next human state and rules. During action selection, the agent selects the joint human–agent action a_t, integrating both human $a_{h,t}$ and agent $a_{m,t}$ actions based on the learned policy. Since the human action $a_{h,t}$ is uncontrollable, the agent determines the agent action $a_{m,t}$ using the joint human–agent action $a_t = \arg\min_{a_{m,t}} f(s_{m,t}, s_{h,t}, s_{d,t})$, where $s_{d,t}$ is the desired state. The agent adds $(s_{m,t}, s_{h,t}, a_t)$ into its replay memory to update its human model. The proposed solution has been shown to achieve a higher cumulative reward.

2) Requesting for Human Action at Uncertain and Unfavorable Circumstances: The agent requests human actions $a_{h,t}$ from human experts in uncertain and unfavorable circumstances. This includes: 1) unusual and unknown events like a robot colliding with obstacles [33], [48], [60], [81] and the system security risk increases [47]; 2) unreliable actions; and 3) actions with rewards below a predefined threshold [81]. During human intervention, the agent observes state s_t, executes human action $a_{h,t}$, and receives reward $r_{t+1}(s_{t+1})$ and next state s_{t+1}. The experience $(s_t, a_{h,t}, r_{t+1}(s_{t+1}), s_{t+1})$, including requested human actions $a_{h,t}$, can be: 1) stored for training and action selection [81] and 2) used to refine human models (see Section II-E) for generating more appropriate human actions. The proposed solution has been shown to achieve a higher cumulative reward [81], [82].

3) Using Human Inputs to Select a Subset of Possible Actions: The agent can select appropriate actions or remove inappropriate ones from the action space based on human inputs [9], which can be positive or negative human rewards [9]. This helps to avoid inappropriate next state while ensuring not being conservative in action selection. Actions with human rewards below a threshold are removed. In [35], the agent uses a shield to select appropriate actions based on human inputs, determining shield parameters with independent probability distributions (e.g., the normal distribution). Stability is improved because significant changes to human inputs do not affect the probability distributions significantly.

In [35], HITL-RL alternates between two loops. The *outer* loop uses: 1) trained policies from the inner loop and 2) human inputs from crowdsourcing. Based on this information, the agent: 1) estimates human perceptions using multiple trained policies and shield parameters; 2) evaluates action appropriateness; and 3) updates shield parameters over time. The shield parameter θ_i follows a probability distribution, such as a normal distribution with mean μ_i and variance σ_i, adjusted based on human inputs. The *outer* loop updates the shield parameters for the inner loop. The *inner* loop uses Bayesian updates and the outer loop's shield parameters to provide its own shield parameters, selecting a subset of appropriate actions $A_{\text{app}}(s, \theta) := \cap_{i=1}^{N_\psi} \psi_i(s, \theta_i) \subseteq A$ from multiple shields N_ψ, each with a shield parameter distribution θ_i, and learns the best policy over time. The inner loop then provides the trained policy to the outer loop.

The proposed solution has been shown to reduce the frequency of requests for human inputs, improving convergence rate, cumulative reward, and task success rate.

C. Guided Exploration

Guided exploration uses human inputs to guide an agent in exploration to increase its convergence rate. Traditional RL explores the entire state and action spaces, including unlikely optimal actions, though the optimal state–action pair is only a small part. The challenge is minimizing exploration time in complex high-dimensional spaces with varying relevance and importance in various parts of the spaces.

1) Using Minimal Human Inputs to Guide State Space Exploration: Human inputs efficiently guide state exploration. In [61], humans provide contextual information $\mathcal{C}$, representing a small set of states $S^* = [\bar{s}_1, \ldots, \bar{s}_K]$ relevant to the important parts of the state space. These diversified states cover various scenarios for better generalization, including inaccurate and suboptimal ones for exploration. The agent employs discriminators; each ϕ_l determines whether a state is relevant or not using a CNN-based encoder $f_{\text{enc}}(\cdot)$. These discriminators maximize an objective function $\max_{\phi_l} E_{S^*}[\log(\phi_l(f_{\text{enc}}(s^*)))] + E_s[\log(1 - \phi_l(f_{\text{enc}}(s)))]$, where the first part is the expected log-likelihood of observing the relevant state s^* under the discriminator ϕ_l given the distribution S^*, and the second part is a regularization term. The agent then learns an optimal exploration policy $a_t \sim \pi_{\text{exp}}(\cdot|s_t, \mathcal{C})$ to guide exploration toward relevant parts of the state space, maximizing the cumulative exploratory reward $\max_{\pi_{\text{exp}}} E_{a_t \sim \pi_{\text{exp}}(\cdot|s_t, C)}[r_{t+1}(s_{t+1}), \mathcal{C}]$. Achieving a higher exploratory reward indicates that the current state is closer to the relevant states given by humans. The proposed algorithm has been shown to achieve a higher interaction count with relevant states and system performance, and reduces human effort.

2) Using Human Inputs to Guide Action Space Exploration: Human inputs efficiently guide action selection. In [83], humans select agent actions $a_m = a_h \in A$ aligned with their preferences, guiding exploration. Humans select actions a_h based on past experiences and also learn over time. The proposed solution has been shown to achieve a higher accumulated rewards and reduced agent failures to complete processes within predefined limits, frequency of random actions selected for a state, and the number of steps needed for each process.

D. Augmented Value Function

The augmented value function uses human inputs to update an agent's the learning function (e.g., Q-function) [31].

1) Using Human Inputs to Adjust Loss Function: Human inputs as scalar values adjust the loss function in DL. In [11], humans provide two types of feedback on the policy applied in complex high-dimensional states during training. First, *evaluative feedback* represents the appropriateness of a selected agent action in a state: 1) $q_t(s_t, a_t) = +1$ for appropriate actions; 2) $q_t(s_t, a_t) = 0$ for inappropriate actions that *cannot* be corrected by human; and 3) $q_t(s_t, a_t) = \beta$ for inappropriate actions that *can* be corrected by human, where β prioritizes corrected samples based on the current ratio of non-corrected to corrected samples [84]. Second, the *corrective feedback* provides direction for correcting inappropriate agent actions through $a_t \leftarrow a_t + correction$. Corrected actions are executed and stored in replay memory for training.

Using an asynchronous algorithm (see Section II-F1), the agent trains with a batch of corrected samples $(s_t, a_t, q_t(s_t, a_t))$ from replay memory, reinforcing appropriate actions for each state. Using DRL, the agent minimizes the gradient of the loss function, which represents the negative gradient of the policy's log-likelihood $\nabla_\theta L(s_t, a_t) = -q_t(s_t, a_t)\nabla_\theta \log(\pi_\theta(a_t|s_t))$, to update its network parameters θ, reducing exploration, and training time and samples needed for training. The proposed solution has been shown to achieve a higher task execution success rate.

2) Learning Reward Values Through Demonstration: IRL enables an agent to learn a complex reward function from human demonstrations (comprised of state–action pairs) and feedback (based on reward parameters and state features).

Most IRL algorithms store human demonstrations in replay memory for training [85], [86]. In contrast, in [39], HITL-IRL, based on a deep neural network, reduces human involvement given human demonstrations in two ways [39]. First, complex tasks (goals) are segregated into a sequence of subtasks (subgoals). The agent starts from the initial state, learns or performs a subtask, and progresses to the next one when the current one performs well until it reaches the final state. Human involvement is minimized by providing demonstrations for subtasks that the agent struggles to complete within a given time period. Humans provide demonstrations for significant states commonly involved in multiple tasks and replace insignificant states with alternatives to enhance the reward function. Second, the agent adjusts the reward function for training neural network parameters θ_d and θ_f by maximizing the difference between the feature expectation and failure experience and matching between the feature expectation and successful demonstration, specifically $r_t(s_t) = g(\theta_d, s_t) + \theta_f \cdot g_{fc1}(\theta_d, s_t)$, where s_t is the state feature, $g(\cdot)$ is the reward function of the state feature, and $g_{fc1}(\cdot)$ is the second last layer output vector. The proposed

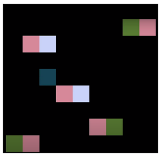


Fig. 6. Asterix screenshot. Dark blue box represents the player, pink/white boxes represent treasures, and pink/green boxes represent enemies.

solution has been shown to reduce the number of steps needed to complete a task.

V. Comparison of the Concept Designs of HITL-RL

This preliminary study investigates the concept designs in an Atari game, providing insights into future research directions in this new research topic. Atari games have been widely served as proxies for complex human-centered environments in RL algorithm investigations [28], [87].

A. Objectives of the Investigation

While human knowledge has been shown to improve the HITL-RL performance [87], our focus is to compare the three separate concept designs (HI-AI, AI-HI, and parallel-HI-and-AI), which have not been investigated in the literature. The two main investigations are the convergence property and system performance, focusing on the instantaneous scores and their standard deviation.

B. Models of Concept Designs for Atari Games

System-level investigations, such as human–machine frameworks, typically assess performance across applications [10], while algorithm-level investigations assess algorithmic performance in dynamic environments like Atari games [88], [89]. This article evaluates the concept designs using the Atari game Asterix, as shown in Fig. 6 [90]. In Asterix, a player moves up, down, left, right, or stays still at each decision epoch. Treasures and enemies spawn from the screen's sides. The player earns a score of 1 by picking up treasures (pink and white boxes). The game starts with the player at the center and ends when they touch an enemy (pink and green boxes). Difficulty increases as the spawn rate and speed of treasures and enemies rise. The HITL-RL model for the three concept designs has three representations. The state is $s_t = (s_t^{x_i}, s_t^{y_i}, s^{n_i}) \in S, i \in \{1, \ldots, i, \ldots, I\}$, where $s_t^{x_i}, s_t^{y_i} \in \{1, 2, \ldots, 10\}$ are x and y positions of an entity n_i, and $s^{n_i} \in \{\text{player}, \text{treasure}, \text{enemy}\}$ is one of the $|I|$ entities. The action $a_t \in A = \{\text{left}, \text{right}, \text{up}, \text{down}, \text{no action}\}$ represents a player's movement at decision epoch t. The reward is $r_{t+1}(s_{t+1}) = +1$ for picking up a treasure; otherwise, $r_{t+1}(s_{t+1}) = 0$.

C. Algorithms of Concept Designs for Atari Games

In HI-AI, the human player's keyboard input is the human action incorporated into the state, extending the state representation to $s_t = (s_t^{x_i}, s_t^{y_i}, s^{n_i}, a_{h_t}) \in S$, $i \in \{1, \ldots, i, \ldots, I\}$. In AI-HI, the human player verifies and replaces inappropriate agent actions shown on the computer screen with a human action if needed by pressing a key on the keyboard. In parallel-HI-and-AI, the human player verifies and replaces inappropriate agent rewards shown on the computer screen with a negative constant reward $r_{t+1}(s_{t+1}) = -1$ if needed.

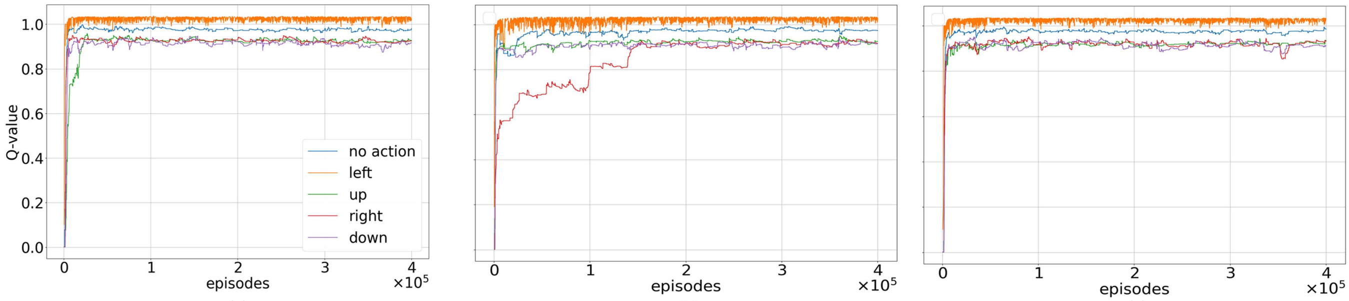
D. Simulation Setup and Parameters

Our simulation environment, based on Python [92] and OpenAI Gym [91], includes codes for the baseline algorithm (Q-learning) and the Asterix environment, available from the public GitHub repository [90]. Five expert human players sit at their computer screens to play the game, well-versed in the game's dynamics and appropriate actions for each state. We conduct 400 000 episodes, each representing a game with varying decision epochs. Human involvement is limited to the initial 450 decision epochs, lasting approximately 20 min at the normal pace of gameplay, due to typically short duration (see Table I). The learning rate is $\alpha = 0.1$ and the discount factor is $\gamma = 0.95$. The exploration probability is decayed from an initial value of $\varepsilon = 1$ at time $t_s = 1$ to $\varepsilon = \varepsilon/(t_e - t_s)$ at time $t_e = 200\,000$. The human player observes the current state on the computer screen and inputs human actions via keyboard keys (i.e., left, right, up, down, or space for no action) to provide human inputs. The Q-values and scores are recorded. This project has received ethical approval from Universiti Tunku Abdul Rahman, with informed consent obtained from all human players involved.

E. Simulation Results and Discussion

Fig. 7 demonstrates the convergence of Q-values of the possible actions for the most popular state across three AuI approaches, where Q-values for all possible actions exhibit minimal change between episodes. HI-AI converges in 15 000 episodes, AI-HI converges in 150 000 episodes, and parallel-HI-and-AI converges in 24 000 episodes. The low convergence rate of AI-HI is not observed in other states, explaining that human inputs are subjective. In AI-HI, the "moving right" action was less frequently selected, resulting in a slower convergence rate. However, convergence is still evident in other states, emphasizing that human involvement maintains RL convergence, which is important to ensure that the agent achieves its optimal policy.

Fig. 8 indicates similar average instantaneous scores for HITL-RL and Q-learning. However, parallel-HI-and-AI outperforms AI-HI and HI-AI. This is because humans replace the agent reward with a negative constant $r_{t+1}(s_{t+1}) = -1$ when the agent actions are inappropriate in the current state. The negative rewards deter inappropriate actions, resulting in faster learning. In contrast, AI-HI and HI-AI lack such negative reinforcement, resulting in slower learning. During initialization, agents' unfamiliarity with the environment leads to negative scores in the first 450 decision epochs, reducing Q-values for inappropriate actions and facilitating faster learning, a feature absent in AI-HI and HI-AI approaches where there is a lack of negative rewards and slower learning. The only time an agent receives a negative reward is when it encounters an enemy.



(a)

(b)

(c)

Fig. 7. Q-values of the possible actions of the most popular state for the HI-AI, AI-HI, and parallel-HI-and-AI algorithms converge. Each graph is plotted based on 400 000 episodes played by a single human player. (a) HI-AI. (b) AI-HI. (c) Parallel-HI-and-AI.

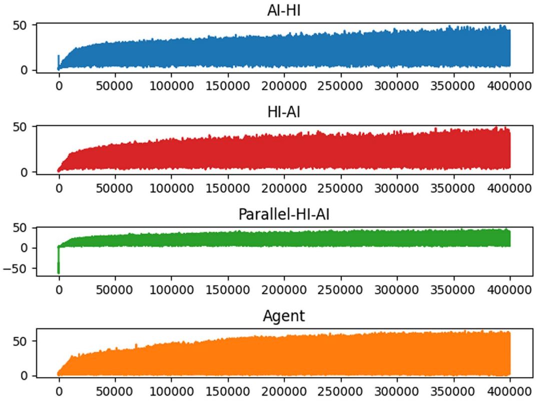


Fig. 8. Average instantaneous scores for AI-HI, HI-AI, parallel-HI-and-AI, and Q-learning (or Agent) algorithms are similar. Each result represents the average instantaneous scores achieved by the five human players.

TABLE V
STANDARD DEVIATION OF SCORES ACHIEVED BY THE AI-HI, HI-AI, PARALLEL-HI-AND-AI, AND Q-LEARNING ALGORITHMS

AI-HI	HI-AI	Parallel-HI-and-AI	Q-learning (Agent)
6.32	6.40	6.00	13.20

Table V indicates that HITL-RL has a significantly lower standard deviation compared to Q-learning, demonstrating higher human involvement and system stability.

F. Summary

This section presents a preliminary study on AuI-based concept designs and traditional Q-learning in an Atari game, providing insights into future directions in this new research topic. Human involvement maintains RL convergence, ensuring that the agent achieves its optimal policy. AI-HI, HI-AI, and parallel-HI-and-AI improve system stability while achieving approximately similar average instantaneous scores to traditional Q-learning. However, several factors require further investigation: 1) the number of human players and episodes with subjective human inputs; 2) state and action space sizes; 3) human expertise and the time to become experts; and 4) simulation parameters, including learning rate α, discount factor γ, and exploration probability ε.

VI. Future Research Directions

This section presents research gaps in this topic.

A. Processing Human Inputs and Outputs of HITL-RL

Human inputs and outputs, like perception, are subjective and nonmeasurable, whereas agent inputs and outputs, such as temperature, are objective and measurable. Modules can be designed for ease of use in HITL systems (see Section II-F3), improving human willingness to interact with the systems. There are four types of modules to explore in HITL-RL. First, preprocessors: 1) incorporate subjective nonmeasurable human inputs, such as business acumen and human touch into states; 2) remove human biases; and 3) clean, filter, and analyze human datasets. Second, HI capitalizers leverage HI to: 1) include new states, actions, and learning objectives flexibly; 2) reduce interactions with the environment; and 3) provide lightweight solutions. Third, action processors adjust agent action based on human judgments and values. Fourth, reward processors modify human rewards to reflect satisfaction with state–action pairs.

B. Addressing the Dynamics of Human Behaviors and Needs

Human inputs are dynamic, yet traditional RL often assumes static human behavior. Also, HITL approaches assume that human inputs are accurate and precise, neglecting the influence of internal factors (e.g., feelings) and external factors (e.g., environment). Dynamic human behaviors and needs can change optimal agent actions, convergence rates, and learning goals.

There are five research gaps.

1) *Addressing dynamic state and action spaces:* Traditional RL and HITL-RL assume static state and action spaces. Human behaviors' dynamics necessitate the inclusion of new states and actions, exploring their impacts [93].
2) *Adjusting agent dependence on human inputs:* Human behaviors' dynamics influence an agent's reliance on human inputs, shifting the balance between human and agent involvement. This fluctuation changes system's inclination toward manual or automated operation.
3) *Determining interaction points between humans and agents:* Each interaction between a human and an agent generates an experience. While more interactions aid

agent learning, they increase human effort. Sufficient experiences allow reduced human involvement. On the other hand, when experiences are insufficient, a task can be decomposed into smaller parts, each handled by a single agent. This increases interaction points and experiences as multiple agents collect their respective experiences [93].

4) *Determining proactive or reactive human involvement:* In proactive involvement, humans decide when to engage in training and decision making with the agent [11], whereas, in reactive involvement, humans participate only when queried by the agent. Choosing the suitable approach or a blend of both ensures timely human involvement across diverse applications.

5) *Addressing irreproducible experimental results with subjective nonmeasurable human inputs:* HITL-RL experiments may lack reproducibility due to the subjective nature of human inputs. New methodologies accommodating this factor can be developed.

C. Catering to Human Limitations

Addressing human limitations enhances the human experience. Imposing high expectations on human inputs may lead to frustration, affecting human–agent interactions. There are three research gaps.

1) *Addressing human response time:* Humans require longer response times than agents, posing a bottleneck in real-time systems. Response time should be adjusted based on the current action's positive and negative effects, balancing the risk of delayed action with human comfort. Striking this balance avoids frustration while ensuring timely responses.

2) *Smoothing human–agent collaboration:* Transitioning control between a human and an agent can lead to confusion, especially in passive tasks with sporadic human inputs. When control shifts to a human, unexpected changes, loss of attention, comprehension time, and adjustment difficulties may arise [70]. Conversely, when control shifts to an agent, humans may feel uncomfortable with the abrupt authority change. Smoother collaboration alleviate these challenges.

3) *Acknowledging diverse time requirements for humans to become experts:* Initially, humans are nonexperts, with transition periods to expertise influenced by individual capabilities, intelligence, interest, and attitude. Unlike agents, humans' expertise acquisition time varies, necessitating consideration of these factors when evaluating time requirements to become experts.

D. Ensuring the Trustworthiness of Human Inputs and Outputs of HITL-RL

The trustworthiness of HITL-RL relies on human perception and behavior. Humans may consciously or unconsciously bias states and actions to serve their interests, manipulating rewards for personal gain. Higher trustworthiness increases the quality of human inputs, reducing human involvement time. To enhance HITL-RL trustworthiness, focus on: 1) verifying humans and their inputs and 2) cleaning and filtering human inputs.

E. Combining Different AuI and HITL Approaches and Their Extensions

Typically, existing approaches employ a single HITL approach. However, in [51], human involvement in both agent inputs (i.e., reward) and outputs (i.e., action) yielded higher average rewards than solely influencing agent outputs. Exploring combinations of various AuI (e.g., HI-AI, AI-HI, and parallel-HI-and-AI) and HITL approaches (e.g., reward shaping, policy shaping, guided exploration, and augmented value function) warrants further investigation.

F. Exploring the Use of Concept Designs in Various D(RL) Algorithms and Applications

The concept designs are versatile and applicable across diverse D(RL) algorithms and applications involving human inputs. For extending D(RL) algorithms, traditional RL and HITL-RL are typically single-agent and centralized. They can expand into a multiagent distributed algorithm, enabling multiple agents to share knowledge and learn collaboratively without interference among neighboring agents. This fosters collaboration among multiple humans and multiple agents toward a common goal, despite subjective differences in individual opinions. For extending applications, concept designs enhance human–machine interaction: 1) improving customer relationships and ultimately boosting marketing performance, including conversion and retention rates, in digital marketing; and 2) improving data collection by autonomous aerial vehicles (AAVs) from multiple mobile users and ultimately boosting application performance and reducing algorithmic complexity in AAV applications [94]. The concept designs can be integrated into human–machine frameworks [10], which have been tested in applications like person identification and video anomaly detection [10], to further enhance their performance.

VII. Conclusion

AuI integrates human inputs into AI, while HITL-RL integrates human inputs into traditional RL. Though sharing similarities, the literature has not explored the relationship between AuI and HITL-RL. From the AuI perspective, this article identifies three HITL-RL concept designs that harness both HI and AI to leverage their respective strengths. The three separate designs are: 1) HI-AI, which includes human actions in the agent state; 2) AI-HI, where humans verify agent actions and select human actions; and 3) parallel-HI-and-AI that combines human and agent states, actions, rewards, or a mix. A preliminary study on the three HITL-RL concept designs demonstrates that human involvement maintains RL convergence, with the AI-HI, HI-AI, and parallel-HI-and-AI approaches enhancing system stability in an Atari game. This article addresses a timely topic as both HITL-RL and AuI are in their early stages. Future research directions include addressing the dynamics of human behaviors and needs and catering to human limitations.

REFERENCES

[1] K.-L. A. Yau et al., "Augmented intelligence: Surveys of literature and expert opinion to understand relations between human intelligence and artificial intelligence," *IEEE Access*, vol. 9, pp. 136744–136761, 2021.

[2] V. G. Cerf, "Augmented intelligence," *IEEE Internet Comput.*, vol. 17, no. 5, pp. 96–96, Sep./Oct. 2013.

[3] S. Liu, S. Zhao, Y. Pang, and Z. Chen, "Human machine joint decision making in distorted surveillance scenario," in *Proc. 2nd China Symp. Cogn. Comput. Hybrid Intell.*, Xi'an, China, 2019, pp. 47–52.

[4] S. Wang et al., "Fusion of machine intelligence and human intelligence for colonic polyp detection in CT colonography," in *Proc. IEEE Int. Symp. Biomed. Imag.: Nano Macro*, 2011, pp. 160–164.

[5] L. Huanghuang, L. Yang, H. Cheng, W. Tu, and M. Xu, "Human-in-the-loop reinforcement learning," in *Proc. Chin. Automat. Congr.*, Jinan, China, 2017, pp. 4511–4518.

[6] N. Alamdari, E. Lobarinas, and N. Kehtarnavaz, "Personalization of hearing aid compression by human-in-the-loop deep reinforcement learning," *IEEE Access*, vol. 8, pp. 203503–203515, 2020.

[7] C. Li, P. Zheng, S. Li, Y. Pang, and C. K. M. Lee, "AR-assisted digital twin-enabled robot collaborative manufacturing system with human-in-the-loop," *Robot. Comput.-Integr. Manuf.*, vol. 76, 2022, Art. no. 102321.

[8] H. (H.) Huang, J. Si, A. Brandt, and M. Li, "Taking both sides: Seeking symbiosis between intelligent prostheses and human motor control during locomotion," *Curr. Opin. Biomed. Eng.*, vol. 20, pp. 1–6, 2021.

[9] K. Blanchet, A. Bouzeghoub, S. Kchir, and O. Lebec, "How to guide humans towards skills improvement in physical human-robot collaboration using reinforcement learning?," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2020, pp. 4281–4287.

[10] Z. Yu, Q. Li, F. Yang, and B. Guo, "Human-machine computing," *CCF Trans. Pervasive Comput. Interact.*, vol. 3, pp. 1–12, 2021.

[11] E. Chisari, T. Welschehold, J. Boedecker, W. Burgard, and A. Valada, "Correct me if i am wrong: Interactive learning for robotic manipulation," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3695–3702, Apr. 2022.

[12] A. C. Tenorio-Gonzalez, E. F. Morales, and L. Villasenor-Pineda, "Dynamic reward shaping: Training a robot by voice," in *Proc. 12th Ibero-Amer. Conf. AI*, Bahía Blanca, Argentina, 2010, pp. 483–492.

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

[14] V. Mnih et al., "Playing Atari with deep reinforcement learning," in *Proc. NIPS Deep Learn. Workshop*, 2013, pp. 1–9.

[15] C. Nam, P. Walker, M. Lewis, and K. Sycara, "Predicting trust in human control of swarms via inverse reinforcement learning," in *Proc. IEEE Int. Symp. Robot Hum. Interact. Commun.*, Lisbon, Portugal, 2017, pp. 528–533.

[16] A. Holzinger, "Interactive machine learning for health informatics: When do we need the human-in-the-loop?," *Brain Informat.*, vol. 3, pp. 119–131, 2016.

[17] X. Wu, L. Xiao, Y. Sun, J. Zhang, T. Ma, and L. He, "A survey of human-in-the-loop for machine learning," *Future Gener. Comput. Syst.*, vol. 135, pp. 364–381, 2022.

[18] L. C. da Cruz, C. A. Sierra-Franco, G. F. M. Silva-Calpa, and A. B. Raposo, "Enabling autonomous medical image data annotation: A human-in-the-loop reinforcement learning approach," in *Proc. 16th Conf. Comput. Sci. Intell. Syst.*, Sofia, Bulgaria, 2021, pp. 271–279.

[19] L. Schiatti, J. Tessadori, N. Deshpande, G. Barresi, L. C. King, and L. S. Mattos, "Human in the loop of robot learning: EEG-based reward signal for target identification and reaching task," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 4473–4480.

[20] J. O. Coelho, L. P. Gaspary, and L. M. R. Tarouco, "How much management is management enough? Providing monitoring processes with online adaptation and learning capability," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag.*, New York, NY, USA, 2009, pp. 299–302.

[21] J. Zhong and Y. Li, "Toward human-in-the-loop PID control based on CACLA reinforcement learning," in *Proc. Int. Conf. Intell. Robot. Appl.*, Shenyang, China, 2019, pp. 605–613.

[22] F. Cicirelli, A. Guerrieri, C. Mastroianni, G. Spezzano, and A. Vinci, "Thermal comfort management leveraging deep reinforcement learning and human-in-the-loop," in *Proc. IEEE Int. Conf. Hum.-Mach. Syst.*, Rome, Italy, 2020, pp. 1–6.

[23] T. Kim and J.-H. Lee, "C-3PO: Cyclic-three-phase optimization for human-robot motion retargeting based on reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 8425–8432.

[24] S. S. Shuvo and Y. Yilmaz, "CIBECS: Consumer input based electric vehicle charge scheduling for a residential home," in *Proc. North Amer. Power Symp.*, College Station, TX, USA, 2021, pp. 1–6.

[25] M. Maadi, H. A. Khorshidi, and U. Aickelin, "A review on human-AI interaction in machine learning and insights for medical applications," *Int. J. Environ. Res. Public Health*, vol. 18, pp. 1–27, 2021.

[26] G. Warnell, N. Waytowich, V. Lawhern, and P. Stone, "Deep tamer: Interactive agent shaping in high-dimensional state spaces," in *Proc. 32nd AAAI Conf. AI*, 2018, pp. 1545–1553.

[27] S. Elmalaki, "FaiR-IoT: Fairness-aware human-in-the-loop reinforcement learning for harnessing human variability in personalized IoT," in *Proc. Int. Conf. IoT Des. Implementation*, 2021, pp. 119–132.

[28] M. Agarwal, S. K. Venkateswaran, and R. Sivakumar, "Human-in-the-loop RL with an EEG wearable headset: On effective use of brainwaves to accelerate learning," in *Proc. 6th ACM Workshop Wearable Syst. Appl.*, 2020, pp. 25–30.

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

[30] A. Fachantidis, M. Taylor, and I. Vlahavas, "Learning to teach reinforcement learning agents," *Mach. Learn. Knowl. Extraction*, vol. 1, no. 1, pp. 21–42, 2019.

[31] T. Brys, A. Harutyunyan, H. B. Suay, S. Chernova, M. E. Taylor, and A. Nowe, "Reinforcement learning from demonstration through shaping," in *Proc. 24th Int. Conf. AI*, 2015, pp. 3352–3358.

[32] T. Kim and J.-H. Lee, "TeachMe: Three-phase learning framework for robotic motion imitation based on interactive teaching and reinforcement learning," in *Proc. 28th IEEE Int. Conf. Robot Hum. Interact. Commun.*, New Delhi, India, 2019, pp. 1–8.

[33] M. Mainampati and B. Chandrasekaran, "Implementation of human in the loop on the TurtleBot using reinforced learning methods and robot operating system (ROS)," in *Proc. IEEE 12th Annu. Inf. Technol., Electron. Mobile Commun. Conf.*, 2021, pp. 448–452.

[34] Q. Li, Z. Yu, L. Yao, and B. Guo, "RLTIR: Activity-based interactive person identification via reinforcement learning tree," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4464–4475, Mar. 2022.

[35] D. Marta, C. Pek, G. I. Melsión, J. Tumova, and I. Leite, "Human-feedback shield synthesis for perceived safety in deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 406–413, Jan. 2022.

[36] T. Mandel, Y.-E. Liu, E. Brunskill, and Z. Popovic, "Where to add actions in human-in-the-loop reinforcement learning," in *Proc. 31st Conf. AI*, 2017, pp. 2322–2328.

[37] J. Lin, Z. Ma, R. Gomez, K. Nakamura, B. He, and G. Li, "A review on interactive reinforcement learning from human social feedback," *IEEE Access*, vol. 8, pp. 120757–120765, 2020.

[38] P. T. Rajendran, H. Espinoza, A. Delaborde, and C. Mraidha, "Human-in-the-loop learning methods toward safe DL-based autonomous systems: A review," in *Proc. Comput. Saf., Rel. Secur.*, 2021, pp. 251–264.

[39] X. Pan and Y. Shen, "Human-interactive subgoal supervision for efficient inverse reinforcement learning," in *Proc. 17th Int. Conf. Auton. Agents MultiAgent Syst.*, Stockholm, Sweden, 2018, pp. 1380–1387.

[40] D. Zha, K.-H. Lai, M. Wan, and X. Hu, "Meta-AAD: Active anomaly detection with deep reinforcement learning," in *Proc. IEEE Int. Conf. Data Mining*, Sorrento, Italy, 2020, pp. 771–780.

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

[42] G. Gordon et al., "Affective personalization of a social robot tutor for children's second language skills," in *Proc. 30th AAAI Conf.* AI, Phoenix, AZ, USA, 2016, pp. 3951–3957.

[43] M. Sharif, D. Erdogmus, C. Amato, and T. Padir, "End-to-end grasping policies for human-in-the-loop robots via deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, Xi'an, China, 2021, pp. 2768–2774.

[44] X. Gao, J. Si, Y. Wen, M. Li, and H. Huang, "Reinforcement learning control of robotic knee with human-in-the-loop by flexible policy iteration," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5873–5887, Oct. 2022.

[45] B. A. Plummer, M. H. Kiapour, S. Zheng, and R. Piramuthu, "Give me a hint! Navigating image databases using human-in-the-loop feedback," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Waikoloa, HI, USA, 2019, pp. 2048–2057.

[46] F. Cruz, G. I. Parisi, J. Twiefel, and S. Wermter, "Multi-modal integration of dynamic audiovisual patterns for an interactive reinforcement learning scenario," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Daejeon, South Korea, 2016, pp. 759–766.

[47] E. C. Williams, N. Gopalan, M. Rhee, and S. Tellex, "Learning to parse natural language to grounded reward functions with weak supervision," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 1–7.

[48] L. Yang, Q. Sun, N. Zhang, and Z. Liu, "Optimal energy operation strategy for We-Energy of energy internet based on hybrid reinforcement learning with human-in-the-loop," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 32–42, Jan. 2022.

[49] M. Hamaya, T. Matsubara, T. Noda, T. Teramae, and J. Morimoto, "Learning assistive strategies for exoskeleton robots from user-robot physical interaction," *Pattern Recognit. Lett.*, vol. 99, pp. 67–76, 2017.

[50] M. Hamaya, T. Matsubara, T. Noda, T. Teramae, and J. Morimoto, "Learning task-parameterized assistive strategies for exoskeleton robots by multi-task reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, Singapore, 2017, pp. 5907–5912.

[51] W. Saunders, G. Sastry, A. Stuhlmüller, and O. Evans, "Trial without error: Towards safe reinforcement learning via human intervention," in *Proc. 17th Int. Conf. Auton. Agents Multi-Agent Syst.*, Stockholm, Sweden, 2018, pp. 2067–2069.

[52] G. Chalvatzaki, X. S. Papageorgiou, P. Maragos, and C. S. Tzafestas, "Learn to adapt to human walking: A model-based reinforcement learning approach for a robotic assistant rollator," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 3774–3781, Oct. 2019.

[53] 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.

[54] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[55] S. Krening and K. M. feigh, "Interaction algorithm effect on human experience with reinforcement learning," *ACM Trans. Hum.-Robot Interact.*, vol. 7, no. 2, 2018, Art. no. 16.

[56] Y. Vasylkiv et al., "Automating behavior selection for affective telepresence robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, Xi'an, China, 2021, pp. 2026–2032.

[57] L. Tao, M. Bowman, J. Zhang, and X. Zhang, "Forming real-world human-robot cooperation for tasks with general goal," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 762–769, Apr. 2022.

[58] E. Teng and B. Iannucci, "Autonomous curiosity for real-time training onboard robotic agents," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Waikoloa, HI, USA, 2019, pp. 1486–1495.

[59] G. Yu, E. Barut, and C. Su, "Introducing deep reinforcement learning to NLU ranking tasks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Toronto, ON, Canada, 2021, pp. 3465–3469.

[60] B. Chandrasekaran and M. Mainampati, "A human in the loop based robotic system by using soft actor critic with discrete actions," in *Proc. 4th Int. Conf. Mechatronics, Robot. Automat.*, China, 2021, pp. 19–24.

[61] A. S. Chen, H. Nam, S. Nair, and C. Finn, "Batch exploration with examples for scalable robotic reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4401–4408, Jul. 2021.

[62] Y. Wen, A. Brandt, J. Si, and H. H. Huang, "Automatically customizing a powered knee prosthesis with human in the loop using adaptive dynamic programming," in *Proc. Int. Symp. Wearable Robot. Rehabil.*, Houston, TX, USA, 2017, pp. 1–2.

[63] G. Yu, Z. He, C. Lai, and Y. Sun, "An optimization design system with hybrid intelligence," in *Proc. 5th World Congr. Intell. Control Automat.*, Hangzhou, China, 2004, pp. 2790–2794.

[64] K.-J. Wang, C. Y. Zhang, and Z.-H. Mao, "Human-centered, ergonomic wearable device with computer vision augmented intelligence for VR multimodal human-smart home object interaction," in *Proc. 14th ACM/IEEE Int. Conf. Hum.-Robot Interact.*, Daegu, South Korea, 2019, pp. 767–768.

[65] A. Hebbar, "Augmented intelligence: Enhancing human capabilities," in *Proc. 3rd Int. Conf. Res. Comput. Intell. Commun. Netw.*, Kolkata, India, 2017, pp. 251–254.

[66] S. Doltsinis, P. Ferrira, and N. Lobse, "A symbiotic human-machines learning approach for production ramp-up," *IEEE Trans. Hum.-Mach. Syst.*, vol. 48, no. 3, pp. 229–240, Jun. 2018.

[67] B. Maettig and H. Foot, "Approach to improving training of human workers in industrial applications through the use of intelligence augmentation and human-in-the-loop," in *Proc. 15th Int. Conf. Comput. Sci. Educ.*, Delft, The Netherlands, 2020, pp. 283–288.

[68] A. Agrawal et al., "A novel hybrid intelligence approach for 2D packing through internet crowdsourcing," in *Proc. IEEE Technol. Innov. ICT Agriculture Rural Develop.*, Chennai, India, 2015, pp. 33–39.

[69] M. E. Koujok, A. Ragab, H. Ghezzaz, and M. Amazouz, "A multi-agent-based methodology for known and novel faults diagnosis in industrial processes," *IEEE Trans. Ind. Inform.*, vol. 17, no. 5, pp. 3358–3366, May 2021.

[70] M. Sridhar1 and C. Wu, "Piecewise constant policies for human-compatible congestion mitigation," in *Proc. IEEE Intell. Transp. Syst. Conf.*, 2021, pp. 2499–2505.

[71] S. Akbarzadeh, N. Alamdari, C. Campbell, E. Lobarinas, and N. Kehtarnavaz, "Word recognition clinical testing of personalized deep reinforcement learning compression," in *Proc. IEEE 14th Dallas Circuits Syst. Conf.*, Dallas, TX, USA, 2020, pp. 1–2.

[72] Y. Wen, X. Gao, J. Si, A. Brandt, M. Li, and H. H. Huang, "Robotic knee prosthesis real-time control using reinforcement learning with human in the loop," in *Proc. Int. Conf. Cogn. Syst. Signal Process.*, Beijing, China, 2018, pp. 463–473.

[73] Y. Wen, M. Li, J. Si, and H. Huang, "Wearer-prosthesis interaction for symmetrical gait: A study enabled by reinforcement learning prosthesis control," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 4, pp. 904–913, Apr. 2020.

[74] J. Jarrett, M. B. Blake, and I. Saleh, "Crowdsourcing, mixed elastic systems and human-enhanced computing—A survey," *IEEE Trans. Serv. Comput.*, vol. 11, no. 1, pp. 202–214, Jan./Feb. 2018.

[75] G. Kuhlmann, P. Stone, R. Mooney, and J. Shavlik, "Guiding a reinforcement learner with natural language advice: Initial results in RoboCup soccer," in *Proc. 31st Conf. AI*, CA, USA, 2004, pp. 30–35.

[76] R. Maclin and J. W. Shavlik, "Creating advice-taking reinforcement learners," *Mach. Learn.*, vol. 22, nos. 1–3, pp. 251–281, 1996.

[77] A. L. Thomaz and C. Breazeal, "Teachable robots: Understanding human teaching behavior to build more effective robot learners," *Artif. Intell.*, vol. 172, no. 6/7, pp. 716–737, 2008.

[78] W. Chu and Z. Ghahramani, "Preference learning with Gaussian processes," in *Proc. 22nd Int. Conf. Mach. Learn.*, Bonn, Germany, 2005, pp. 137–144.

[79] M. Tucker et al., "Preference-based learning for exoskeleton gait optimization," in *Proc. Int. Conf. Robot. Autom.*, 2020, pp. 2351–2357.

[80] G. Chen, Z. Sabato, and Z. Kong, "Semantic parsing of automobile steering systems," in *Proc. Int. Conf. IoT*, 2018, pp. 1–3.

[81] M. Tarle, M. Bjorkman, M. Larsson, L. Nordstrom, and G. Ingestrom, "A world model based reinforcement learning architecture for autonomous power system control," in *Proc. Int. Conf. Commun. Control Comput. Technol. Smart Grids*, Aachen, Germany, 2021, pp. 364–370.

[82] G. A. Vouros, "Explainable deep reinforcement learning: State of the art and challenges," *ACM Comput. Surv.*, vol. 55, no. 5, pp. 1–39, 2022.

[83] S. Doltsinis, P. Ferreira, and N. Lohse, "A symbiotic human-machine learning approach for production ramp-up," *IEEE Trans. Hum.-Mach. Syst.*, vol. 48, no. 3, pp. 229–240, Jun. 2018.

[84] A. Mandlekar, D. Xu, R. Martìn-Martìn, Y. Zhu, L. Fei-Fei, and S. Savarese, "Human-in-the-loop imitation learning using remote teleoperation," 2020, *arXiv:2012.06733*.

[85] C.-E. Hsu, M. Rohmatillah, and J.-T. Chien, "Multitask generative adversarial imitation learning for multi-domain dialogue system," in *Proc. IEEE Autom. Speech Recognit. Understanding Workshop*, 2021, pp. 954–961.

[86] M. Xu, L. Yang, X. Tao, Y. Duan, and Z. Wang, "Saliency prediction on omnidirectional image with generative adversarial imitation learning," *IEEE Trans. Image Process.*, vol. 30, pp. 2087–2102, 2021.

[87] L. Guan, M. Verma, S. Guo, R. Zhang, and S. Kambhampati, "Widening the pipeline in human-guided reinforcement learning with explanation and context-aware data augmentation," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, pp. 21885-21897.

[88] Z. Gao et al., "Dynamic memory-based curiosity: A bootstrap approach for exploration in reinforcement learning," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 8, no. 2, pp. 1181–1193, Apr. 2024.

[89] Q. Yang, H. Wang, M. Tong, W. Shi, G. Huang, and S. Song, "Leveraging reward consistency for interpretable feature discovery in reinforcement learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 2, pp. 1014–1025, Feb. 2024.

[90] K. Young and T. Tian, "MinAtar: An Atari-inspired testbed for thorough and reproducible reinforcement learning experiments," 2019, *arXiv:1903.03176*.

[91] Gym, Accessed: 8 Sep. 2023. [Online]. Available: https://gym.openai.com/

[92] Python, Accessed: 8 Sep. 2023. [Online]. Available: https://www.python.org/

[93] L. Cai, C. Wu, K. J. Meimandi, and M. S. Gerber, "Adaptive mobile behavior change intervention using reinforcement learning," in *Proc. Int. Conf. Companion Technol.*, Ulm, Germany, 2017, pp. 1–2.

[94] Z. Dai et al., "AoI-minimal UAV crowdsensing by model-based graph convolutional reinforcement learning," in *Proc. IEEE Conf. Comput. Commun.*, London, U.K., 2022, pp. 1029–1038.