HHAI 2023: Augmenting Human Intellect
P. Lukowicz et al. (Eds.)

doi:10.3233/FAIA230098

Reinforcement Learning Requires Human-in-the-Loop Framing and Approaches

Matthew E. TAYLOR
The University of Alberta (matthew.e.taylor@ualberta.ca) &
The Alberta Machine Intelligence Institute (matt.taylor@amii.ca) &
AI-Redefined (matt@ai-r.com)

Abstract. Reinforcement learning (RL) is typically framed as a machine learning paradigm where agents learn to act autonomously in complex environments. This paper argues instead that RL is fundamentally human in the loop (HitL). The reward functions (and other components) of a Markov decision process are defined by humans. The decisions to tackle a certain problem, and deploy a learned solution, are taken by humans. Humans can also play a critical role in providing information to the agent throughout its life cycle to better succeed at the problem in question. We end by highlighting a set of critical HitL research questions, which, if ignored, could cause RL to fail to live up to its full potential.

Keywords. Reinforcement Learning, Human-Agent Interaction, Human in the Loop, Interactive Machine Learning

1. Introduction

Reinforcement learning (RL) is a common approach for training autonomous agents to learn sequential decision-making tasks. While this research has spanned decades, it has recently gained prominence in the research community due to its multiple successes in difficult benchmark tasks. Unfortunately, there have been relatively few instances of successful deployment of RL in real-world tasks.[1] We argue that to begin addressing key questions that enable real-world RL applications ,we should think of RL as a process for integrating and leveraging human knowledge, rather than fully autonomous agent learning.

The typical academic framing of a Markov decision process (MDP), as done in Sutton and Barto [1], shows an agent interacting with an environment (see Figure 1). In contrast with this traditional view of RL, our goal is to argue three points:

1. RL (like most machine learning) is fundamentally human in the loop (HitL);
2. RL agents can greatly benefit from humans (or other agents) — ignoring such existing knowledge typically slows down learning significantly; and

[1]For the purposes of this paper, we will define real-world tasks as ones where the agent makes impactful decisions in the real world. This could be a physical robot performing a cleaning task or a virtual agent taking actions to trade stocks. An agent playing a simulated game without real-world consequences does not qualify.

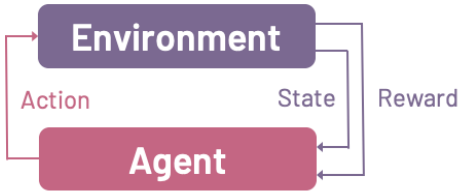

Figure 1. An RL agent can learn to act in a Markov decision process, but this process must be defined by a human. Without specifying the actions, state formulation, and the reward function, an RL agent would not be able to learn.

3. If our community ignores the critical role that humans play in RL, we will miss many opportunities for RL to succeed in impactful settings.

Many argue that just having a reward function is sufficient for agents to learn very complex behaviors:

> In this paper, we consider... that the generic objective of maximising reward is enough to drive behaviour that exhibits most if not all abilities that are studied in natural and artificial intelligence. [2]

While we agree that reward can be used to allow an RL agent to accomplish many different tasks, we argue that this framing may cause researchers to miss a key point: *reinforcement learning is fundamentally a human-in-the-loop paradigm*. To this point, the reward function must come from somewhere — currently, a reward function *must* be defined by a human, not the agent itself. It is possible that RL algorithms may advance to the point where an agent can go out into the real world and learn tasks on its own. However, in order for current methods to solve real-world problems, a human, often a subject matter expert (SME), needs to be intimately involved in the project to achieve successful agent performance.

In order to limit the scope of this paper, we make four assumptions. First, the applications considered are "real-world" and are not toy (benchmark) domains like Atari or board games. Second, people with different skill sets[2] are willing and able to help, even if they are fallible. Third, we only consider single-agent domains (omitting discussions around multi-agent RL and human-agent teaming). Finally, although agents can also help other agents learn, we limit ourselves to external human assistance.

Section 2 of this paper will discuss how humans are necessarily involved in multiple phases of the RL problem formulation, solution, and deployment. Section 3 discusses how potential solutions to open questions raised in Section 2 might be evaluated. Section 4 concludes with a call for the community to recognize the importance of humans in the RL process and to focus time and energy on the problems raised. Ultimately, we hope the reader will agree that the problems raised are indeed critical blockers that could prevent RL from realizing its full potential in real-world problems.

[2]Later, we will explicitly consider how subject matter experts, machine learning developers, computer scientists, and laypeople can help in different phases of the learning process.

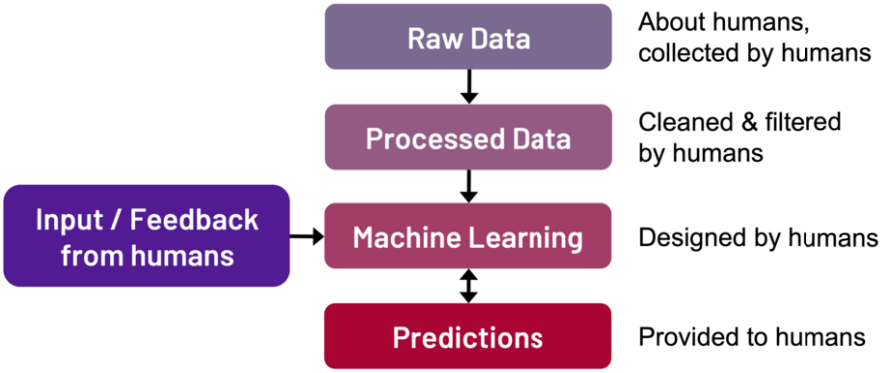

Figure 2. Humans play a critical role in machine learning tasks. This picture is adapted from Figure 1 in Mathewson and Pilarski [3].

Table 1. People with different expertise will be more or less useful at different phases of the RL project life cycle.

	People	Business	SME	DevOps	Laypeople	ML/RL
Problem Phase	1: Identification	x				x
	2: Formulation	x	x			x
	3: Learning		x			x
	3′: Human-Accelerated Learning	x	x	x	x	x
	4: Deployment	x	x	x		x
	5: Maintenance			x		x

2. Five phases of RL development

Others have previously argued that all machine learning (ML) is fundamentally human-in-the-loop. For instance, Mathewson and Pilarski [3] argue that ML is a multi-step process, and each step may rely on human input (see Figure 2). Ultimately, any ML process is: initiated by a human, using data that is often cleaned by a human, running a human-designed algorithm (possibly with the assistance from humans while learning), on a learning objective defined by a human, and by providing predictions a human can act on.

In this paper, we argue for a slightly different framing by focusing on RL and breaking the process into five[3] distinct phases, listed in Figure 3. This section of the paper argues why humans play a critical role in each phase.

Throughout these different phases, people with different types of expertise will be more or less useful, as discussed in the following subsections and summarized in Table 1.[4]

[3]Other divisions of the RL problem formulation and life cycle are possible [4].

[4]Note that we use broad and overlapping categories to describe different types of human knowledge — more nuanced views, including people's level of expertise, should be considered in the future.

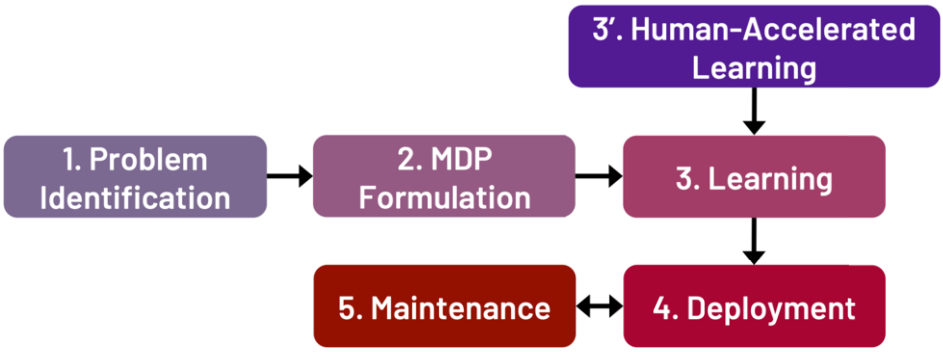

Figure 3. This paper discusses 5 steps towards successful RL deployment, each of which involves humans. For exposition purposes, we differentiate between how humans are necessarily part of the decisions around the learning process (step 3) from how humans could assist in accelerating/improving an agent's learning (step 3′).

2.1. Step 1: Problem Identification

RL is an amazing hammer, but it needs to find the right nail. Applying RL to inappropriate problems is likely to result in failure [5]. In order to identify a problem that RL can potentially solve with an appropriate amount of resources, an evaluation is needed from scientific, business, and subject-specific angles. Current agents can not perform any of these kinds of analyses.

Identifying an appropriate problem requires both technical and business input. From the technical side, one needs to consider whether the problem can be modeled well as an MDP, how difficult it will be for the agent to perform the task, whether RL can outperform an existing process (if any), and what data or simulators are available. Companies need to be convinced that the return on investment (ROI) will be worth the effort by balancing how impactful a successful solution will be, how likely the research team will be successful in training an agent, and how long the team will take to implement a solution. A key open problem is how to best identify potential RL problems in business settings and then make a risk/reward judgement on whether the project is worth pursuing. There are many possible problems that RL could address — it is a general and flexible framework. However, unlike supervised learning, it can be difficult in practice to understand when RL would be effective and worth the upfront cost required to generate a proof of concept or minimal viable product. This is particularly true if the company does not have access to experts with multiple years of RL experience.

Open questions include how to write guidelines to help businesses identify problems that could benefit from RL, how to evaluate their feasibility, and how to estimate their potential ROI.

2.2. Step 2: Formulating the MDP

In most RL research settings, we assume we are provided a defined (and fixed) MDP, while businesses must generally construct an appropriate MDP. The action set, state variables, and reward must be carefully selected or constructed so that the agent is able to learn with a minimal amount of environmental interaction, while still achieving performance high enough to make it worthwhile to deploy.

For example, the action set should not be too large (slower to learn), too small (cannot achieve optimal performance), too low-level (forcing the agent to learn concepts that could be easily provided), or too high-level (not letting the agent to fine tune its behavior). Similarly, the reward function must be carefully constructed so that the agent is able to learn quickly, but also achieve high asymptotic performance on the targeted real-world metric.[5] Improving elicitation techniques would better allow domain experts to assist machine learning technicians in defining the problem.

Open questions include how to best work with SME to (iteratively) elicit knowledge to create and refine an MDP, how to best align the reward function with what the business desires, and how to best find the "Goldilocks Zone" where the MDP is only as detailed as it needs to be (and no more).

2.3. Step 3: Learning

This subsection considers how humans must make decisions before an agent can learn on the defined MDP and the first consideration relates to data. Is there data that can be used to help the agent bootstrap with offline or batch learning methods? If there is no simulator, is it worth constructing a simulator, and how accurate does it need to be? What are the best ways to transfer knowledge from a simulator to an agent acting in the real world (e.g., sim2real [13])? How should one balance between an abstract simulation that is fast to learn and a more realistic simulation that is slower? If no simulator is used, how can the agent learn as quickly as possible while minimizing regret and other potential side effects (e.g., wear and tear on a robot)?

Whether learning in simulation or in the real world, a particular learning approach needs to be selected (e.g., value-based, policy search, or a genetic algorithm) along with a particular algorithm. If a neural network is used, an appropriate architecture needs to be selected and hyperparameters that need to be tuned. While all of these could potentially be chosen automatically (e.g., by auto-ML [14] and neural architecture search [15] methods), such a search is generally quite constrained by the amount of data required (or, if the agent is acting in a purely digital environment, the amount of simulation time required). When selecting how much compute or time to spend on hyperparameter tuning, one must also consider how much total compute or time should be spent before the system can be deployed. Or, if the system must act in the real world, how much compute or time is expected to be required before the agent can outperform an existing system?[6]

Open questions include discovering best practices for when an agent should learn offline vs. learn in a simulator vs. learn online in the real world and how to balance the amount of compute/time between hyperparameter tuning and learning.

[5] We note that there are methods to help address some of these problems. For instance, option discovery [6,7,8] can be used to construct (useful) temporally extended actions, representation learning [9,10] is a general area of research attempting to discover or construct useful features, and appropriate reward shaping [11,12] could help find a more useful reward function. However, we believe that significant algorithmic work is required before these methods can easily help construct an MDP that could outperform one carefully designed by a SME.

[6] For instance, an agent that must control a plant in the real world could be benchmarked against an existing rule-based controller, or could be benchmarked in terms of how much learning is required before the agent can learn a policy that is good enough for the plant to be profitable.

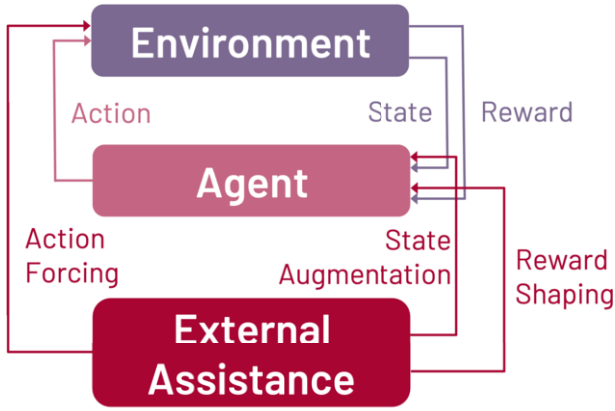

Figure 4. A human has many different ways of influencing an agent's learning before or during learning.

2.4. Step 3′: Human-Accelerated Learning

While the prior section focused on the entire learning process, this section discusses how to best incorporate human knowledge into the learning process itself. Every machine learning algorithm has biases and it may make sense to include such biases from knowledgeable humans in order to speed up the learning process. For instance, one could explicitly consider a human as part of the system and how that human could help the agent best learn — a number of assistance types are highlighted in Figure 4. There are many examples in the literature that show human assistance can significantly improve learning — if such help is available, past research [16,17,18] suggests that it does indeed make sense to leverage it. Furthermore, the previous section argued that humans are involved throughout the RL life cycle — assisting agent learning is just one more place where human knowledge can be leveraged.

How a human can best assist an agent is an open area of research and the goal of this paper is not to exhaustively list the many different ways humans can improve agent learning. Instead, the interested reader is directed to existing surveys on this topic. For example, there are many ways of categorizing the different methods of human assistance, including

- What type of assistance is provided? [16]
- How is the assistance used or interpreted? [16]
- Is the assistance general or specific to a certain state (or set of states)? [17]
- What is the modality of the assistance (e.g., textual vs. gestures vs. pressing buttons)? [17]
- Is the human or agent in control of the overall process? [18]

A human's individual expertise and capabilities determine the types of assistance they should be asked to provide to the agent. For example, a pilot can demonstrate flying a plane (i.e., using a learning from demonstration algorithm [19]), while a layperson could provide useful positive or negative feedback [20] or provide preferences over different trajectories [21].

Open questions include if and how we can develop guidelines about when one type of assistance would be better than another, which will likely depend on factors like

the human's abilities, the task difficulty, and the learning algorithm used; how should different types of advice be combine, even if they conflict; how to best leverage limited human input across large, complex tasks; and what are best practices around providing information to a user (e.g., explainability or transparency of the agent).

2.5. Step 4: Deciding to Deploy

At some point, an agent learning in simulation may be given the go-ahead to start acting in the real world. Similarly, an agent acting in the real world in a test environment may eventually be promoted to production. These decisions critically depend on human assessment and their understanding of the risk/reward tradeoffs of allowing the agent to take actions with real-world consequences.

A closely related question to whether the agent has trained enough is whether the MDP formulation needs to be revisited. For example, a policy that earns high reward may in fact produce unwanted behavior, whether it is a bicycle riding in circles [22] or a boat getting stuck in a reward loop rather than learning to complete the task [23].

Open questions include discovering best practices for evaluating RL agents before deploying to the real world or production, or if they should re-formulate the underlying MDP and try to learn a higher performing policy.

2.6. Step 5: System/Agent Maintenance

During and after the agent is deployed, there will be critical questions about what hardware is required, how to ensure robustness and safety, and what types of real-time human supervision will be required. But even once the agent is successfully running, there are additional considerations related to whether the agent continually learns, if/when the agent needs to re-train, and if/when the problem formulation needs to be revisited. For example, if there is a large change to the transition function due to changes in the system, one might want the agent to re-train from scratch. Or, if there are new inputs available due to additional business knowledge, how should these new features be incorporated into the agent? Or, if the needs of the business changes, how should reward function be changed and how does this affect the agent's learning?

Open questions are less well defined in this setting because there have been relatively few deployed RL systems — as more RL systems are deployed, we will very likely identify novel problems related to safety, reliability, trust, continual learning, and non-stationarity, all of which require HitL thinking to develop best practices. One particularly relevant question is how to best develop systems that are maintainable by companies that do not have RL expertise in house, as this is currently not a common skill in industry.

3. Evaluation of Research Questions

In this work we argue that it is critical to involve humans throughout the RL problem life cycle and highlight open questions. Unfortunately, these research questions are difficult to evaluate, much less benchmark, as is common practice in RL (and ML in general). Instead, we argue we should develop approaches and best practices and evaluate them individually on the different phases of the problem.

A first step would be to come up with proposed best practices, similar to software engineering best practices, for one or more of the problems. For instance, in step 3′, one could propose a set of heuristics for when one type of advice would be more useful than another, and then test these different types of advice over a set of benchmark tasks.

However, because it is difficult to construct a set of benchmark tasks that are likely to represent many different real-world tasks, a more holistic (e.g., a case-study-based design) approach to understanding the likely success of different methods could be undertaken by looking at how RL currently works (or fails) in the real world via case studies. It will be critical for academics who do fundamental RL research to work closely with businesses that apply RL to real world problems. This is necessary to identify gaps in our understand and identify what is most critical to allowing humans to help RL succeed.

4. Conclusion & Call to Action

Our hope is that this paper has challenged the reader to not think of RL as a fully autonomous paradigm, but instead as an iterative learning and development process involving both learning algorithms and humans. While better algorithms may chip away at this assumption, full autonomy in terms of problem identification, construction, and deployment is unlikely in the near-future. Instead, we argue that it is critical to consider how humans and RL agents can work together to solve sequential decision tasks.

Some readers may find the fundamental and applied research questions raised in this paper "obvious." However, we argue that based on the research in our community, these problems are either unrecognized or under-valued. In order for RL to be as impactful as possible in the near- to medium-term, we believe it is critical to tackle these questions head-on. For example, if we always assume the MDP is formulated correctly and provided to the agent, *our community will miss out on important research questions, limiting the impact of RL in the real world*.

In order for the RL community (and the agents community in general) to maximize its impact, it is important to refine and tackle these problems at the interface between humans and RL agents. This will require more interaction between academics working on fundamental research and practitioners who are using RL to solve real-world problems on a daily basis. Workshops series like "RL for Real Life" [24] are an important start, but much more remains to be done. We hope that the reader will consider how their RL (and agent-based) research could better benefit from explicitly considering the human-agent interface in order to maximize the likelihood of successfully deploying RL solutions.

When we successfully develop best practices for how potential users of RL can best formulate useful problems, help RL agents learn to quickly solve impactful tasks, and understand how to move RL agents into production, we will be much closer to turning RL into an impactful solutions method. Ultimately, this will help practitioners no longer think that RL is "Unwieldy and It Takes a Lot of Time" [25] but instead have it come into mainstream usage, outside of academia and on increasingly important real-world problems.

Part of this work has taken place in the Intelligent Robot Learning (IRL) Lab at the University of Alberta, which is supported in part by research grants from the Alberta Machine Intelligence Institute (Amii); a Canada CIFAR AI Chair, Amii; Compute Canada; Huawei; Mitacs; and NSERC. We also thank Bei Peng, Calarina Muslimani, Laura Petrich, Robert Loftin, and Tianpei Yang for their detailed feedback.

References

[1] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018.

[2] Silver D, Singh S, Precup D, Sutton RS. Reward is enough. Artificial Intelligence. 2021;299:103535. Available from: https://www.sciencedirect.com/science/article/pii/S0004370221000862.

[3] Mathewson KW, Pilarski PM. A Brief Guide to Designing and Evaluating Human-Centered Interactive Machine Learning. arXiv; 2022. Available from: https://arxiv.org/abs/2204.09622.

[4] Li P, Thomas J, Wang X, Khalil A, Ahmad A, Inacio R, et al. RLOps: Development Life-Cycle of Reinforcement Learning Aided Open RAN. IEEE Access. 2022;10:113808-26.

[5] Dulac-Arnold G, Mankowitz D, Hester T. Challenges of Real-World Reinforcement Learning; 2019. Available from: https://openreview.net/forum?id=S1xtR52NjN.

[6] Sutton RS, Precup D, Singh S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. Artificial Intelligence. 1999;112(1):181-211. Available from: https://www.sciencedirect.com/science/article/pii/S0004370299000521.

[7] Bacon PL, Harb J, Precup D. The Option-Critic Architecture. Proceedings of the AAAI Conference on Artificial Intelligence. 2017 Feb;31(1). Available from: https://ojs.aaai.org/index.php/AAAI/article/view/10916.

[8] Bagaria A, Konidaris G. Option Discovery using Deep Skill Chaining. In: International Conference on Learning Representations; 2020. Available from: https://openreview.net/forum?id=B1gqipNYwH.

[9] Lange S, Riedmiller M, Voigtlander A. Autonomous reinforcement learning on raw visual input data in a real world application; 2012. p. 1-8.

[10] Stooke A, Lee K, Abbeel P, Laskin M. Decoupling Representation Learning from Reinforcement Learning. In: Meila M, Zhang T, editors. Proceedings of the 38th International Conference on Machine Learning. vol. 139 of Proceedings of Machine Learning Research. PMLR; 2021. p. 9870-9. Available from: https://proceedings.mlr.press/v139/stooke21a.html.

[11] Ng AY, Harada D, Russell SJ. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In: Proceedings of the Sixteenth International Conference on Machine Learning. ICML '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1999. p. 278–287.

[12] Hu Y, Wang W, Jia H, Wang Y, Chen Y, Hao J, et al. Learning to Utilize Shaping Rewards: A New Approach of Reward Shaping. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS'20. Red Hook, NY, USA: Curran Associates Inc.; 2020. .

[13] Tobin J, Fong R, Ray A, Schneider J, Zaremba W, Abbeel P. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE Press; 2017. p. 23–30. Available from: https://doi.org/10.1109/IROS.2017.8202133.

[14] He X, Zhao K, Chu X. AutoML: A survey of the state-of-the-art. Knowledge-Based Systems. 2021;212:106622. Available from: https://www.sciencedirect.com/science/article/pii/S0950705120307516.

[15] Liu H, Simonyan K, Yang Y. DARTS: Differentiable Architecture Search; 2018. Cite arxiv:1806.09055. Available from: http://arxiv.org/abs/1806.09055.

[16] Bignold A, Cruz F, Taylor ME, Brys T, Dazeley R, Vamplew P, et al. A conceptual framework for externally-influenced agents: an assisted reinforcement learning review. J Ambient Intelligent Human Computation. 2021 1. Available from: https://dro.deakin.edu.au/articles/journal_contribution/A_conceptual_framework_for_externally-influenced_agents_an_assisted_reinforcement_learning_review/20644473.

[17] Najar A, Chetouani M. Reinforcement Learning With Human Advice: A Survey. Frontiers in Robotics and AI. 2021;8. Available from: https://www.frontiersin.org/articles/10.3389/frobt.2021.584075.

[18] Mosqueira-Rey E, Hernández-Pereira E, Alonso-Ríos D, Bobes-Bascarán J, Fernández-Leal A. Human-in-the-loop machine learning: a state of the art. Artificial Intelligence Review. 2022 08.

[19] Argall BD, Chernova S, Veloso M, Browning B. A survey of robot learning from demonstration. Robotics and Autonomous Systems. 2009;57(5):469-83. Available from: http://www.sciencedirect.com/science/article/pii/S0921889008001772.

[20] Knox WB, Stone P. Reinforcement Learning from Simultaneous Human and MDP Reward. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 1.

AAMAS '12. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems; 2012. p. 475–482.

[21] Christiano PF, Leike J, Brown T, Martic M, Legg S, Amodei D. Deep Reinforcement Learning from Human Preferences. In: Guyon I, Luxburg UV, Bengio S, Wallach H, Fergus R, Vishwanathan S, et al., editors. Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc.; 2017. Available from: https://proceedings.neurips.cc/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.

[22] Randløv J, Alstrøm P. Learning to Drive a Bicycle Using Reinforcement Learning and Shaping. In: Proceedings of the Fifteenth International Conference on Machine Learning. ICML '98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.; 1998. p. 463–471.

[23] Clark J, Amodei D. Faulty Reward Functions in the Wild; 2016 [Online]. OpenAI's blog. Available from: https://openai.com/blog/faulty-reward-functions/.

[24] Li Y, Brunskill E, Chen M, Gottesman O, Li L, Liu Y, et al.. Reinforcement Learning for Real Life (RL4RealLife) Workshop at NeurIPS; 2022. Available from: https://nips.cc/virtual/2022/workshop/50014.

[25] Jacob M, Devlin S, Hofmann K. "It's unwieldy and it takes a lot of time"—Challenges and opportunities for creating agents in commercial games. In: Proceedings of the 16th AAAI international conference on Artificial Intelligence and Interactive Digital Entertainment; 2020. p. 88-94.