Human Influence in the Lifelong Reinforcement Learning Loop

1st Thierry Jacquin
NAVER LABS EUROPE
Meylan, France
thierry.jacquin@naverlabs.com

2nd Julien Perez
NAVER LABS EUROPE
Meylan, France
julien.perez@naverlabs.com

3rd Cécile Boulard
NAVER LABS EUROPE
Meylan, France
cecile.boulard@naverlabs.com

***Abstract*—Deploying embodied agents in real-world settings brings safety, adaptability, and cost challenges. As reinforcement learning (RL) in robotics remains expensive in such scenarios, large scale decision models are often trained with large amount of simulations. Then, transferring to a real-world environment becomes the starting point for a lifelong adaptation learning procedure. In this paper, we introduce a novel learning paradigm that improves the efficiency of adaptation to real environment thanks to an original integration of human expert feedback that iteratively improves the agent's behavior toward an expected one.**

Our approach involves a computational mechanism for which experts may alter the behavior of the agent in a space of influences, that do not need to be aligned with the action space of the targeted environment. It differs from state-of-the-art approaches where the human only intervenes as an oracle over the action space of the target environment. We illustrate our approach with the task of simultaneous localization and mapping and present preliminary results in the context of collision avoidance using the AI-Habitat simulator.

I. Introduction

Embodied agent deployments in public areas emphasize the issue of robot adaptation to ever-changing environments. As an agent, we refer to the sequential decision making model that defines, at each step, the action of the robot. Autonomous robots raise the question of the cost of deployment, the risk of sharing the physical environment with humans, and the limited predictability that robot behavior can provide. Another major concern is the ability to adapt the robots either to a new place or within the same environment to evolving practices and goals. This situation brings the need for lifelong learning for the robot and this concern is shared by researchers from diverse research communities studying the role of humans in the large variety of machine learning paradigms and associated protocols [8]. This general concern was recently labeled human-centered AI [13]. In this paper, we introduce a novel approach to adapting a sequential decision model giving humans the ability to adapt agents for deployment in real-world settings. Interacting with a human in the loop during training has shown significant advantages in numerous real-world situations. First, it allows for improving the data efficiency of the learning process. Second, it makes possible the introduction of robots in new environments in a safe manner [18]. In the following of this paper, we describe a new formalization of humans in the loop where human experts influence the behavior of a sequential decision agent to adapt it to fit their needs. This approach can be distinguished from state-of-the-art approaches of human-in-the-loop where the human is solely defined as oracle over the targeted action space of the considered environment.

We assume that one or several human experts can coach an agent that was newly deployed in the real world or that is facing unforeseen changes in its environment after deployment. As human experts, we refer here to humans that are knowledgeable regarding the task to be done, the environment where the agent is evolving, and the associated risks. No requirements of expertise in algorithms, or models of sequential decision making are assumed. We evidence the technological feasibility of this new learning paradigm with an experiment of autonomous localization and mapping and detail initial results.

II. Human-in-the-loop in Reinforcement Learning

Besides the performance of modern sequential decision learning approaches since Deep Reinforcement Learning [10], [12], recent research works addressing real-world situations where humans interact during such a learning process exist.

First, DQN-TAMER [3] uses face recognition of a supervising human for helping to induce a policy for maze navigation. In this work, the system uses a camera to perceive human faces and interpret them as human feedback using a deep neural network for facial expression recognition. The recognition model is a convolutional neural network-based (CNN) model that classifies facial expressions into 8 categories: 'neutral', 'anger', 'contempt', 'disgust', 'fear', 'happy', 'sadness', 'surprise'. The agent interprets the facial expression 'happy' as positive (+1) and other expressions ('anger', 'contempt', 'disgust', 'fear', and 'sad') as negative (-1). As one possible extension, this study shows a good example of the necessity for humans to help the learning agent using natural language. Nonetheless, while the face recognition approach is an interesting reward signal to use, it is limited to very simple environments and an action frequency that is low. As an alternative, we propose to introduce a task-specific signal of supervision that will directly bias the behavioral policy.

In more complex environments and tasks of sequential decision making, one has first to identify in which parts of the state space to interact in the Human-in-the-Loop learning process. As an example, [11] shows that it is desirable to

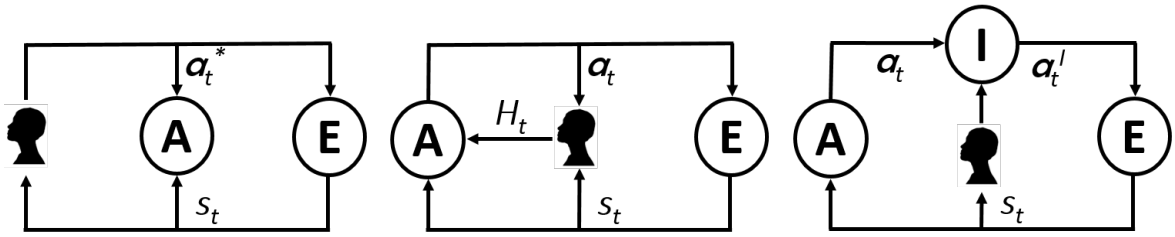

Fig. 1. From left to right: classic imitation learning scheme, evaluative feedback and our proposed coaching approach involving human-in-the-loop sequential learning. The black head is the human expert, also known as coach, E is the environment, and A denotes the agent, a_t is the action chosen at time t by the agent and a_t^* the action expected by the human expert, s_t is the state observed at time t. H_t is an additional reward to maximize provided by the expert regarding agent decisions. In our approach, I is the influence module that transforms the *influence signal* produced by the coach and the agent action into the *influenced action*, a_t^I .

directly interact with the learning agent during its execution if the situation requires it. In our work, we get inspired by this approach, we propose to improve it by letting the human experts decide where and how in the agent environment state space they want to position their advice to create the most impact.

In the specific context of autonomous driving, this question has been addressed by the live supervision of a human driver [17]. It seems natural to supervise autonomous driving with a driver's live corrections as a reinforcement signal. In this context, the shared physical environment forms a convenient situation to align human preferences with the agent decision model. In our work, we improve this scheme when the task cannot be supervised in real-time by a driver, but by regular external observations of a task by experts and without such an action space alignment constraint through the proposed mechanism of influence. Agent-Agnostic Human-in-the-Loop Reinforcement Learning [1] provides an interesting overview of the various algorithmic possibilities offered by the two possible interventions in this learning procedure: altering the action or changing the reward. Action alteration aims at reducing the exploration/exploitation necessity of learning by leveraging human knowledge about the task at hand.

Finally, reward shaping [7] is another approach to learning efficiency improvement that is available at learning time. Action alteration, in contrast, is also available at exploitation time.

III. Influence for human in the loop of sequential decision

Figure 1 describes our proposed method using the notation introduced in [18] and details how we differentiate from it. In comparison to the state-of-the-art methods which mainly consist in either directly fine-tuning the actions of the agent or altering its reward function from feedback provided by the human trainer, our method introduces a conversion step that receives as input both the current action chosen by the agent and an environment-specific influence signal provided by the trainer and produces as output an *influenced* action to execute in the environment.

Given a trained sequential decision policy, our approach is decomposed into two steps. First, the outputs of the policy are altered with constraints, called influence, specified by the human experts and conditioned by the current action and observed state of the environment. In this first adaptation step, where the agent's policy remains unchanged, the model's performance can be degraded. The second step integrates the established influence into the actual model using a straightforward scheme of reward shaping.

In the context of autonomous navigation, an influence can be defined as a direction with an associated magnitude. This magnitude of an influence can skew or smooth the action distribution computed by the influenced agent. Assuming a discrete action space, influences can be applied to the action distribution outputted by the considered policy with an application-specific transformation. As an example, we can alter such policy-selected action distribution by computing the dot-product between it and an action distribution estimated by the influence signal and the associated transformation. Finally, one can select the decision to transmit to the environment by maximizing such *influenced* action distribution. After introducing preliminary notations, we detail a two-step process defining and leveraging this influence signal.

A. Preliminaries

We operate under the assumption that the considered system is described by a Markov Decision Process (MDP) [16]. An MDP consists of the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$ composed of states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, unknown transition dynamics $p(s_{t+1}|s_t, a_t) \sim \mathcal{P}$, a reward function $r \in \mathbb{R}$, and a discount factor $\gamma \in [0, 1]$. At each timestep t, an agent observes the current state s_t, selects an action a_t from a policy $a_t \sim \pi(\cdot|s_t)$, and then observes the reward r_t and the next state s_{t+1}. Solving an MDP consists of finding an optimal policy π^* which maximizes the long-term accumulation of rewards.

B. Influence for action alteration

First, the agent's policy is altered using a task-specific *behavioral function* $\mathbf{B}(\psi, a_t, s_t) \in \mathcal{A}$ which transforms a task-specific influence signal ψ provided by a coach, the current state s_t and currently chosen action a_t into an influenced action, altering the current action. This first step can be tightly associated with the family of approaches of policy shielding [2]. The influence signal ψ and the associated transformation

Fig. 2. Original mapping and navigation (**Left** 88 ops) under right-side influence (**Middle** 24 ops). **Right:** Resulting influenced policy. The trajectory of the SLAM agent is indicated and the color indicates the orientation of the agent while traveling. Red is south, Black is north, Pale green is east, and purple is west oriented.

function $\mathbf{B}(.)$ are task and environment dependent and known by the coach. Various forms of influence signals can be mentioned such as natural language, direct action space, etc. In addition, an influence signal can be conditioned by the current state of the environment or a currently chosen action so the behavioral function takes these information as inputs, too. This alteration of actions is meant for validating, by the coach, the behavior of the influenced policy with respect to the task and the environment at hand before adapting it using a fine-tuning procedure.

C. Learning from influence

In this second step of our approach, the influences defined by the coaches are used to fine-tune the influenced policy. While behavior-cloning [16] could have been considered to fit the policy into the influenced policy defined during the former action alteration step, compound error commonly associated to this approach makes it impractical [6]. Instead, we define a reward-shaping function to reinforce as follows:

$$\mathbf{R}(a_t, s_t, r_t, \psi) = r_t - \lambda\phi(a_t, \mathbf{B}(\psi, a_t, s_t)) \in \mathbb{R} \qquad (1)$$

with ϕ, a reward granted for having chosen an action under the preference distribution computed for the state s_t under the influence and $\lambda \in \mathbb{R}$ is a hyper-parameter. We propose to define such influenced action compliance function as following:

$$\phi(a_t, \mathbf{B}(\psi, a_t, s_t)) = ||a_t - \mathbf{B}(\psi, a_t, s_t)||_2^2, \qquad (2)$$

using the L^2 norm of the difference between the action derived from the influence signal $\mathbf{B}(.)$ and the action chosen by the policy a_t. Reward shaping is well-suited in this context as the agent is encouraged to align with the influence's action without strictly copying it. In this way, the policy can ensure a trade-off between influence compliance and task-specific reward maximization.

This second step allows to decouple the two parts of the approach. First, the interactive adaptation of the policy executed in a given environment, performed during the first step of the approach. Second, a learning procedure as fine-tuning of the current policy once the influenced policy is validated by the coach. Indeed, once the shielded policy is safe and confirmed to adapt to the new context, it can learn the task using state-of-the-art approaches of reward-shaping, defined using the constraints given by the human observer during the first step of the proposed adaptation approach.

IV. Experiments

We illustrate our approach using a simple scenario of autonomous mapping in an indoor environment.

Context and settings: The environment is set as follows, a fleet of robots autonomously navigate in a set of scenes. One of them maps the target area by creating an occupancy grid using SLAM, the others move freely with in the scenes, e.g., executing delivery tasks. It might happen that the mapping robot falsely classifies the delivery robots as static objects and, as a consequence, creates a degraded occupancy grid. This usually happens after a collision with the mapping robot. Thus, the goal of this experiment is to create an influence that helps the mapping robot to better execute its task. On the one hand, we want to minimise the number of collisions, and on the other hand, we want to maximise the size of the successfully mapped areas of the scenes. As simulation framework we use Habitat AI, the target scenes (40) come from the Gibson dataset [14], and we use [5] for SLAM. A collision is detected if the mapping robot gets closer than 0.5m to another robot while driving in opposite directions. More precisely, we assume a deep neural network pre-trained for Simultaneous Localization and Mapping sharing the space with moving obstacles like a fleet of delivery robots to update them with map modifications and also humans. In this experiment, we use the pre-trained neural SLAM model and associated hyper-parameters introduced in [5]. Our study focuses on this navigation behavior part of the problem of the neural SLAM agent. We evaluate

the capability of our approach to influence the robot to adopt a right-side walk influence within the AI-habitat validation dataset while maintaining a functional mapping behavior.

Influences for right-hand side navigation: We define an influence function for adapting the SLAM robot's navigation policy. More precisely, we influence the trained *Global Planner* of the proposed neural architecture that sets the next waypoint in $\mathbb{R}^2$ to reach in the neural navigation stack. We invite the reader to refer to [5] for a complete description of the navigation stack this experiment is based on. In this experiment, we define an influence as a circular displacement of a clock-wise 15-degree angle of the current position of the next waypoint with respect to the robot's current position. As the purpose of this experiment is to prevent collisions, the influence needs to be defined in the reference frame of the robot. So this influence scheme depends on the current global planner action, i.e. the position of the next goal to reach, and the state of the robot, i.e. its current position. The aim of this simple influence is to encourage the navigation policy of the mapper to remain right-sided to prevent collisions with each other. As a performance measure, the collisions are detected using the AI-habitat simulations based on the SLAM robot position and a threshold distance of 0.5 meter. For the sake of simplicity of the simulation, the experiments are performed using only the SLAM agent and collisions are defined as intersections of consecutive positions in the robot trajectory with opposite direction.

Evaluation on step-1, Task compliance: During this first step of the method, the coach influences then validates the navigation behavior of the agents from collisions by defining the influence signal described beforehand. Here, the coach observes the agent behavior and evaluates its appropriateness to the environment. In this illustrative context, the coach monitors the collision rate of the mapper and defines the influence signal that modifies the agent behavior interactively. Figure 2 illustrates the SLAM robot trajectories. While the *safe* mapper maps less completely, it navigates more on the right of the corridors, lowering the risk of frontal collisions. In the resulting mapping, the grey zones on the top left correspond to areas which have not been mapped by the SLAM agent under influence. This lack of mapping is due to the influenced action that prevent the pre-trained SLAM policy to operate as done during its initial training. Figure 3 details the collisions observed for the original, influenced, and fine-tuned policies defined below. Regarding collision avoidance, the influenced policy already reduce the undesirable behaviors of collision. After this first step of influence definition and expected behavior validation, we need to fine-tune the mapper in order to maintain its collision avoidance behavior while improving its mapping performance that has been naturally degraded.

Evaluation on step-2, Reinforcement over influence: In this second step, we set $\lambda = 1.0$ of Equation 1. The reward shaping function defined in Equation 2 is the Euclidian distance between the waypoint computed by the pre-trained global planner of the SLAM agent and the influenced one.

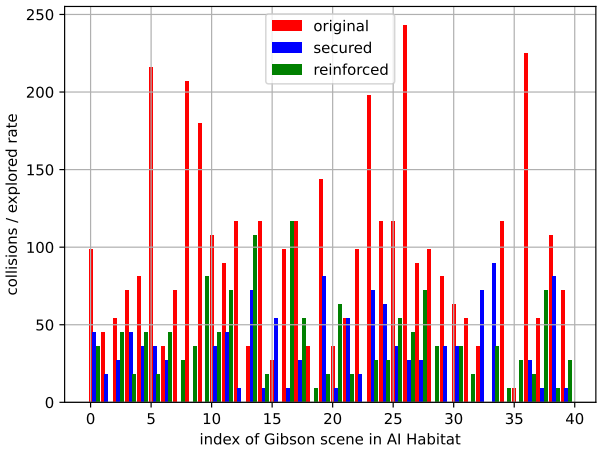

Fig. 3. Collision rates over 40 indoor environments for the task of mapping using the original policy, the secured policy that corresponds to step-1 and finally the reinforced policy produced during step-2 of our approach, the lower the better.

We use the Proximal Policy Optimization algorithm [15] to reinforce the original global planner policy. As illustrated in Figure 3, the navigation model acquires a safer behavior, limiting the registered number of collisions during navigation in all environments. After this reward-shaped reinforcement step based on both the output of the influenced policy defined in step-1 and the original mapping reward of the task, we validate that the mapping correctness, expressed in the surface of correctly estimated space of the map, remains convergent for an equivalent amount of navigation steps with respect to the original policy while limiting the amount of collisions detected by the simulator.

V. Conclusion and Future Work

In this paper, we present a learning paradigm that introduces a novel interaction signal defined as influence, provided by human experts, as a feasible proposition to answer several challenges regarding continuous improvement of policies to changing environments. Our proposition aims at evolving the human interaction from a *targeted task*, that is known to be optimized, to a preferred behavior, that does not need to be fully anticipated by the human experts. Compared to state-of-the-art approaches, our protocol makes human-in-the-loop to shift from being an omniscient oracle to an actor involved in an iterative process of policy improvement. This shift brings two advantages. First, it creates the context for human experts and embodied agents to jointly fine-tune an appropriate behavior. Second, it provides a solution where human experts do not need prior knowledge about algorithms or models of sequential decision making to improve how this technology can support them.

This *blurring the line between developers and end-users* is identified as a critical point to foster wide adoption of AI-based technology [4]. This initial framework presented

in this contribution opens exciting research questions. One open question relates to how human expert influences will be collected. The influence signals can rely on various modalities such as haptic, graphical user interface or natural conversations [3].

As a perspective, we foresee a diversity of human experts ranging from the workers involved in the supervision and maintenance of embodied agents to individual that are physically closed to a robot executing its task. The influence results in the aggregation of the feedback from these human experts, as proposed in [9]. One last element to investigate is the information that we will need to share with the human expert to provide efficient coaching. Each coach will have their representation of the robot's behavior. Complementing this representation with details on the task of the embodied agent, its perception of the environment and its current state might impact the pertinence of the feedback producing the influence signal.

REFERENCES

[1] David Abel, John Salvatier, Andreas Stuhlmüller, and Owain Evans. Agent-agnostic human-in-the-loop reinforcement learning. *arXiv preprint arXiv:1701.04079*, 2017.

[2] Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *AAAI*, 2018.

[3] Riku Arakawa, Sosuke Kobayashi, Yuya Unno, Yuta Tsuboi, and Shin-ichi Maeda. Dqn-tamer: Human-in-the-loop reinforcement learning with intractable feedback. *arXiv preprint arXiv:1810.11748*, 2018.

[4] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[5] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

[6] Peter R. Florence, Corey Lynch, Andy Zeng, Oscar Ramirez, Ayzaan Wahid, Laura Downs, Adrian S. Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *CoRL*, 2021.

[7] Luíza C. Garaffa, Maik Basso, Andréa Aparecida Konzen, and Edison Pignaton de Freitas. Reinforcement learning for mobile robotics exploration: A survey. *IEEE transactions on neural networks and learning systems*, PP, 2021.

[8] Bahar Irfan, Aditi Ramachandran, Samuel Spaulding, Sinan Kalkan, German I Parisi, and Hatice Gunes. Lifelong learning and personalization in long-term human-robot interaction (leap-hri). In *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, pages 724–727, 2021.

[9] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, et al. Webuildai: Participatory framework for algorithmic governance. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW):1–35, 2019.

[10] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[11] Travis Mandel, Yun-En Liu, Emma Brunskill, and Zoran Popović. Where to add actions in human-in-the-loop reinforcement learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[12] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[13] Michael Muller, Plamen Agelov, Shion Guha, Marina Kogan, Gina Neff, Nuria Oliver, Manuel Gomez Rodriguez, and Adrian Weller. Neurips 2021 workshop proposal: Human centered ai.

[14] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019.

[15] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347, 2017.

[16] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[17] Jingda Wu, Zhiyu Huang, Chao Huang, Zhongxu Hu, Peng Hang, Yang Xing, and Chen Lv. Human-in-the-loop deep reinforcement learning with application to autonomous driving. *arXiv preprint arXiv:2104.07246*, 2021.

[18] Ruohan Zhang, Faraz Torabi, L. Guan, Dana H. Ballard, and Peter Stone. Leveraging human guidance for deep reinforcement learning tasks. In *IJCAI*, 2019.