Leveraging Empowerment to Model Tool Use in Reinforcement Learning

Faizan Rasheed[1], Daniel Polani[1], and Nicola Catenacci Volpi[1]

Abstract— **Intrinsic motivation plays a key role in learning how to use tools, a fundamental aspect of human cultural evolution and child development that remains largely unexplored within the context of Reinforcement Learning (RL). This paper introduces "object empowerment" as a novel concept within this realm, showing its role as information-theoretic intrinsic motivation that underpins tool discovery and usage. Using empowerment, we propose a new general framework to model the utilization of tools within RL. We explore how maximizing empowerment can expedite the RL of tasks involving tools, highlighting its capacity to solve the challenge posed by sparse reward signals. By employing object empowerment as an intrinsically motivated regulariser, we guide the RL agent in simple grid-worlds towards states beneficial for learning how to master tools for efficient task completion. We will show how object empowerment can be used to measure and compare the effectiveness of different tools in handling an object. Our findings indicate efficient strategies to learn tool use and insights into the integration and modeling of tool control in the context of RL.**

I. Introduction

Since the advent of the Paleolithic age, tool usage has been a crucial catalyst for the success of human cultural evolution [1]. A wealth of archaeological evidence attests to early humans' innate drive towards tool discovery and control, leading to the development of increasingly sophisticated tools and a complex repertoire at mankind's disposal. This inherent interest in tool usage is not confined to adults; it is observably manifested in the early stages of human child development. Children demonstrate an innate curiosity about objects' functionality and their potential for environmental interaction [2], reinforcing the perception that environmental control through specialized instruments and utensils is a distinctive hallmark of human nature. However, the intrinsic motivation (IM) underpinning humans' persistent drive towards tool discovery, crafting, and usage remains an intriguing area of exploration. With each newly developed tool enhancing the performance of existing skills and introducing a fresh spectrum of interaction possibilities, tool usage essentially expands human capabilities and activities. Given tools' primary objective of augmenting environmental control and diversifying an agent's possibilities, *empowerment* [3] [12] emerges as an IM signal capturing these notions. Therefore, this paper posits empowerment as a natural IM underlying human tool usage and aims to validate this hypothesis by demonstrating how empowerment maximization can accelerate the Reinforcement Learning (RL) of tasks involving tool usage.

Empowerment, an information-theoretic mechanism of IM, measures the extent of influence an agent exercises over its environment. Essentially, empowerment quantifies the repertoire of reliably achievable future options perceivable to the agent [3]. Classically, empowerment is utilized to quantify an agent's impact on its entire environment. However, in this paper, we introduce the concept of "object empowerment". This newly proposed measure quantifies the influence an agent exerts over a specific object in the environment. This is the object which the agent must manipulate (using a tool) to accomplish the assigned task. We intend to illustrate how object empowerment can be leveraged to compare different tools and determine their relative impact on a given object. Additionally, we apply empowerment as an intrinsically motivated regulariser to steer a RL agent towards states advantageous for learning tool usage and subsequently, task completion.

Despite its undeniable importance, the integration and modeling of tool usage within the RL context remains largely unexplored. This paper contributes to fill this gap by investigating the definition and efficient utilization of tools within the RL paradigm. In order to empower an agent to equip and adeptly use a tool, we must address a notorious challenge pervasive in many RL tasks: the sparsity of reward signals [4]. In fact, the benefits of tool usage only become evident upon mastering the necessary control sequence. Unfortunately, these benefits are not discernible through the reward signal prior to the completion of the learning process. As a result, the agent might require extensive environmental exploration to gather the rewards needed for learning how to control the tool. Recognized as a promising solution to the challenge of reward signal scarcity in RL contexts, empowerment has proven its efficacy as an intrinsically motivated regulariser enhancing exploration [5]. This paper seeks to propose and analyze such strategies, contributing to a deeper understanding of tool usage within the RL framework. By doing so, it also aims to shed a brighter light on the the underlying processes behind learning and discovery of tools in general.

II. Related Work

The concept of tool affordances has been a topic of interest in fields such as robotics and artificial intelligence, with several approaches proposed to represent and learn these affordances. Tool affordances can be understood as

*This work was not supported by any organization

[1] The authors are with the Adaptive Systems Research Group, School of Physics, Engineering, and Computer Science, University of Hertfordshire, Hatfield, AL10 9AB UK, f.rasheed@herts.ac.uk

the potential actions that an agent can perform with a tool. Sinapov et al. [6] proposed a behavior-grounded description of tool affordances, where agent learns the effects of their actions with a tool on the environment, a notion that we quantify here with empowerment. The authors report an experiment where a robot learns to represent the tool during a behavioral babbling stage. Jain et al. [7] proposed a strategy where a robot learns to manipulate a target object using a tool through a process of exploration and interaction with the environment, similar to the RL method used here. The robot collects relational instances, which are associations between actions, tools, and effects, and uses these instances to learn probabilistic dependencies within a Bayesian network model. Gonçalves et al. [8] presented a novel computational model of multi-object affordances using Bayesian networks. They consider actions performed using an intermediate affordance (i.e., the tool) to interact with another one (i.e., the object), which recalls the tool-object interactions explored in our framework. These approaches proposed methods that acquire tool affordances with a focus on learning through interaction, exploration and probabilistic modeling, all concepts that our work builds upon.

The process of learning to use tools in RL introduces several challenges, including the need to understand the properties of different tools, the ability to predict their effects and the capacity to control them. IM has emerged as a promising approach to address these challenges. Seepanomwan et al. [9] used IM to enable a humanoid robot to learn new motor skills and their outcomes before exploiting a goal-based mechanism for achieving extrinsic goals. This approach allowed the robot to learn to manipulate a ball on a table using a tool. In [10], the authors proceeded to combine IMs and planning to study the development of tool use. Their agent showed increasingly powerful planning abilities for composing the action sequences needed to solve the tool-using task. Forestier et al. [11] proposed a framework based on IM to explore multiple goals. Therein, the data collected during the exploration of one goal (e.g., mastering a tool) provides information to aid in reaching other goals (e.g., exploiting the tool for other tasks), addressing one of the prerequisites needed for tool usage. These studies demonstrate that IM can drive the exploration and acquisition of new skills, facilitate the understanding of the effects of tool use, and support the planning of action sequences involving tools.

Empowerment [3][12], a well-established IM signal, has proven its utility in scenarios where reward feedback is sparse [5][13]. However, our work aims to take a step forward by studying empowerment in environments where the agent interacts with objects through the use of tools. The idea that empowerment optimization leads agents to interact with objects has been a focal point since early research on empowerment [3]. More recent studies have combined empowerment, along with related information-theoretic quantities, with reinforcement learning to enable robotic manipulators to interact with objects in a self-supervised manner. In this context, a distinction was made between the state of the agent and the state of its surrounding environment. In this regard, [14] aims at maximizing the mutual information between these two variables, where [15] employed the empowerment of the surrounding environment's state. Our work, rather than considering the surrounding environment in general, focuses on the empowerment of the state of a specific object in the environment. This leads to a new instance of empowerment that can be tailored to different objects in the world. This approach could potentially enable a more nuanced and effective interaction between the agent and its environment, in particular in the case where the agents interacts with objects using tools.

III. Methodology

A. Background

1) Reinforcement Learning: RL [16] is a sub-field of machine learning wherein an autonomous agent learns to make decisions through continuous interactions with its environment. The agent engages in a sequential decision-making process, where it takes action based on observed states, procures feedback in the form of rewards or penalties, and refines its strategies to maximize cumulative rewards over an extended period of time. From a formal perspective, RL is modeled as a Markov Decision Process (MDP). The MDP framework is defined by a quadruple $(\mathcal{S}, \mathcal{A}, T, R)$, where $\mathcal{S}$ is the state space, encompassing all possible states an agent can encounter; $\mathcal{A}$ denotes the action space, comprising all possible actions the agent can perform; $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ represents the transition matrix, indicating the probability that an action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$ leads to a state $s' \in \mathcal{S}$; $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function, which represents the immediate reward after transitioning from state s to state s', given that action a was performed. The overarching objective of RL is to ascertain an optimal behaviour that maximizes the expected cumulative discounted reward.

RL often grapples with the 'sparse reward problem', where infrequent extrinsic rewards cause the agent to explore aimlessly in the state-action space due to a lack of frequent feedback. To mitigate this, the concept of *intrinsic motivation* was introduced [17]. Drawing inspiration from developmental psychology and neuroscience [18], IM is typically formalized through the introduction of an auxiliary reward function, augmenting the agent's extrinsic reward signal and encouraging it to explore and understand its environment beyond the pursuit of extrinsic rewards, thus fostering more efficient and robust learning. The resulting regularised reward function $\hat{R}$ can take the following general form

$$\hat{R} \coloneqq R + \beta M \quad , \tag{1}$$

with M denoting the employed IM signal and β the trade-off parameter used to tune the importance of the IM versus the extrinsic reward.

2) Empowerment: Information theory has played a central role in the design of successful IM signals [19]. On of these signals is empowerment [3][12], which is the capacity of an agent's *actuation channel* - i.e., the Shannon channel capacity

[20] of the external part of the action-perception channel. Let us represent an action sequence of length h as $a^h := a_1 a_2 a_3 \ldots a_h \in \mathcal{A}^h$, with $\mathcal{A}^h$ constituting the collection of all possible action sequences of such length. Note that for empowerment calculation, the channel capacity is computed only for *potential* action sequences; these are not action sequences actually taken, but only sequences that *could be* taken in the future. Let us also introduce a given a sensor model $P(O|S)$, where $S \in \mathcal{S}$ is the state random variable, $O \in \mathcal{O}$ is the observation random variable and $\mathcal{O}$ denotes the set of all possible observations. The sensor model indicates the probability distribution of perceiving an observation $o \in \mathcal{O}$ given that the agent is in state s. Given that the agent is in state s at time t, the 'h-step actuation channel' is defined as the triple $\big(A_t^h, P(O_{t+h}|A_t^h, S_t = s), O_{t+h}\big)$. The source of this channel, the random variable $A_t^h \in \mathcal{A}^h$, embodies the potential action sequences of length h that begin in state s at time t. The receiving end of this channel is O_{t+h}, which represents the observation sensed by the agent after h steps. The channel's conditional probability distribution $P(O_{t+h}|A_t^h, S_t = s)$ represents the stochastic relationship between its input and output. Thus, the *h-step empowerment* of state s is defined by the Shannon capacity of the h-step actuation channel, represented as $\mathfrak{E}^h(s) := \max_{P(a^h|s)} I(O_{t+h}; A_t^h|S_t = s)$, where $I(X;Y)$ stands for the mutual information between the random variables X and Y. As a Shannon channel capacity, empowerment is quantified in bits. Furthermore, it operates under an open-loop scheme, meaning that the potential action sequences a^h are operated without utilizing feedback from the environment. In the case of deterministic transition and sensor models, the computation of empowerment reduces to counting the number of different final observations the agent can perceive by modifying the state of the environment via all action sequences of length h [12]. In particular, let us denote with $\mathcal{O}^h(s)$ the set of different observations that the agent can sense by executing all the actions sequences a^h starting in state s. Then, empowerment can be computed by

$$\mathfrak{E}^h(s) = \log_2(|\mathcal{O}^h(s)|) \tag{2}$$

where we used the notation $|\,.\,|$ to indicate the cardinality of a set. We will say that empowerment is *fully observable*, and denote it with $\mathfrak{E}_S^h$ if the agent can observe its full state.

B. Tool Learning Framework

In this section we combine the RL and empowerment formalisms to design a general framework that can be used to model tool learning and characterise tool usage in different environments.

1) State Space: Let us assume that there is an affordance in the environment, named "*tool*", that can be used by the agent to interact with another affordance, which we call "*object*". Let us also assume that, by equipping the tool, the agent can interact more effectively with the object than without it. The state space of the MDP proposed to address this scenario can be decomposed as follows

$$\mathcal{S} := \mathcal{S}^{\mathfrak{A}} \times \mathcal{S}^{\mathfrak{T}} \times \mathcal{S}^{\mathfrak{O}} \times \mathcal{S}^{\mathfrak{W}} \tag{3}$$

Here, $\mathcal{S}^{\mathfrak{A}}$ denotes the state of the agent (e.g., its pose, position, etc.), $\mathcal{S}^{\mathfrak{T}}$ the state of the tool, $\mathcal{S}^{\mathfrak{O}}$ the state of the object and $\mathcal{S}^{\mathfrak{W}}$ the state of any other remaining components of the environment.

2) Action Space: Among all the actions in $\mathcal{A}$ that an agent can perform, we are interested in the subset of actions that have the property of changing the state of the object $s^{\mathfrak{O}} \in \mathcal{S}^{\mathfrak{O}}$. The agent can do this in two ways: either by using the actions that also change the agent state $s^{\mathfrak{A}}$ but not the state of the tool $s^{\mathfrak{T}} \in \mathcal{S}^{\mathfrak{T}}$, which we denote as $\mathcal{A}^{\mathfrak{AO}} \subseteq \mathcal{A}$ and include the possible actions employed by the agent when this has no tool equipped; or by using the actions that also change the state of the tool $s^{\mathfrak{T}}$ once the latter is equipped by the agent, which we denote as $\mathcal{A}^{\mathfrak{TO}} \subseteq \mathcal{A}$.

3) Object Empowerment: We introduce now a novel instance of the empowerment formalism that has an interesting interpretation in the context of our tool learning framework. Let us first define an *"object sensor"* that given a state $s \in \mathcal{S}$ perceives only the corresponding state of the object $s^{\mathfrak{O}} \in \mathcal{S}^{\mathfrak{O}}$ within s. The observation space of the object sensor $\hat{\mathcal{O}}$ is equal to the object's state space $\mathcal{S}^{\mathfrak{O}}$ (i.e., $\hat{\mathcal{O}} = \mathcal{S}^{\mathfrak{O}}$). Then, given a state $s = s^{\mathfrak{A}} \times s^{\mathfrak{T}} \times s^{\mathfrak{O}} \times s^{\mathfrak{W}}$, the object sensor model is defined by $P(\hat{o}|s) := \delta_{\hat{o},s^{\mathfrak{O}}}$, with δ denoting the Kronecker delta function, which is 1 when the considered observation $\hat{o}$ is equal to the object state $s^{\mathfrak{O}}$ of state s, and 0 otherwise. We will denote with *"object empowerment"* the empowerment of the actuation channel that employs the object sensor. The maximisation of object empowerment pushes the agent towards the state with the largest impact on the state of the object. This can be done via either the agent's actions $\mathcal{A}^{\mathfrak{AO}}$ or via the tool's actions $\mathcal{A}^{\mathfrak{TO}}$. Let us define the "agent's object empowerment" $\mathfrak{E}^h_{\mathfrak{AO}}$ as the capacity of the actuation channel $({\mathcal{A}_t^{\mathfrak{AO}}}^h, P(\hat{O}_{t+h}|{\mathcal{A}_t^{\mathfrak{AO}}}^h, S_t = s), \hat{O}_{t+h})$, with input the h-step sequences of agent's actions ${\mathcal{A}_t^{\mathfrak{AO}}}^h$ triggered in state s, and output the state of the object $\hat{O}_{t+h}$ perceived by the agent after h steps. Let us also define the "tool's object empowerment" $\mathfrak{E}^h_{\mathfrak{TO}}$ as the capacity of the actuation channel $({\mathcal{A}_t^{\mathfrak{TO}}}^h, P(\hat{O}_{t+h}|{\mathcal{A}_t^{\mathfrak{TO}}}^h, S_t = s), \hat{O}_{t+h})$, with input the h-step sequences of tool's actions ${\mathcal{A}_t^{\mathfrak{TO}}}^h$ executed in s, and output the state of the object $\hat{O}_{t+h}$ sensed by the agent after h steps. The prior assumption, which posits that the agent must be capable of interacting with the object when using the tool, can be formalized by the condition $\mathfrak{E}^h_{\mathfrak{TO}} > 0$. Furthermore, $\mathfrak{E}^h_{\mathfrak{TO}}$ can be used to quantify how much the tool is effective in influencing the state of the object. In principle, different pairs of tools and objects could be considered and compared with respect to the magnitude of this quantity. To express the condition that a tool is indeed useful, we could write $\mathfrak{E}^h_{\mathfrak{AO}} < \mathfrak{E}^h_{\mathfrak{TO}}$; if that condition does not hold, it means that the tool actually encumbers the agent with respect to object manipulation. In other words, when this condition is satisfied, by using the tool the agent has more impact on the object than by not using it. Note how these statements can be expressed both with respect to a single state $s \in \mathcal{S}$, using the per-state empowerment values $\mathfrak{E}^h_{\mathfrak{A}\mathfrak{O}}(s)$ and $\mathfrak{E}^h_{\mathfrak{T}\mathfrak{O}}(s)$, or in terms of the whole MDP, averaging empowerment over an uniform distribution of states, yielding average empowerment values $\hat{\mathfrak{E}}^h_{\mathfrak{A}\mathfrak{O}}$ and $\hat{\mathfrak{E}}^h_{\mathfrak{T}\mathfrak{O}}$.

IV. Experiments

In this section, we numerically investigate the role of fully observable empowerment (FOE) $\mathfrak{E}^h_S$ and tool's object empowerment (TOE) $\mathfrak{E}^h_{\mathfrak{T}\mathfrak{O}}$ for learning of tool usage in two experiments involving two simple grid-world environments.

A. Experimental Setup

In the following sections, we consider two different 10 x 10 grid-world-like MDPs (Fig. 3a and Fig. 4a). Let us denote by $\mathcal{W}$ all possible cells in the grid. The grid-worlds' states $s \in \mathcal{S}$ can be decomposed in three components: the agent state $s^{\mathfrak{A}} \in \mathcal{W}$, which in the figure is represented by a robot, and indicates the agent position in the grid; the tool state $s^{\mathfrak{T}} \in \mathcal{W}$, which is depicted as either a picker or a broom, depending on which of these two tools is present in the environment, and denotes the tool's location in the world; the object state $s^{\mathfrak{O}} \in \mathcal{W}$, which is represented as a can and indicates the object's location in the grid. Furthermore, additional components of the environment include the goal state $s_g \in \mathcal{W}$, which is depicted as a waste bin, and the wall states, which obstruct the agent navigation and are represented by black cells.

The employed action set is defined as $\mathcal{A} = \{\uparrow_A, \rightarrow_A, \downarrow_A, \leftarrow_A, \uparrow_T, \rightarrow_T, \downarrow_T, \leftarrow_T\} = \mathcal{A}_A \cup \mathcal{A}_T$. The actions $a_A \in \mathcal{A}_A$ denote the possible one-cell movements that the agent can do towards the cardinal directions of north, east, south, and west. Using a tool action $a_T \in \mathcal{A}_T$ only has an effect if the tool is equipped. In this case, a_T moves the tool to the location relative to the agent indicated by its direction (see Fig. 1a). Let us denote by d_M the Manhattan distance between two cells in the grid-world. We will say that a tool is "equipped" by the agent if the first is adjacent to the latter (i.e., $d_M(s^{\mathfrak{T}}, s^{\mathfrak{A}}) = 1$). The equipment happens automatically when the agent is located in a cell adjacent to a tool, without using any specific action. Note that once the tool is equipped by the agent, it will remain equipped until the end of the episode. This means that whenever the agent moves, it also "carries" the tool with it (see Fig. 1b and 1c for examples about how agent movements impact the tools' positions).

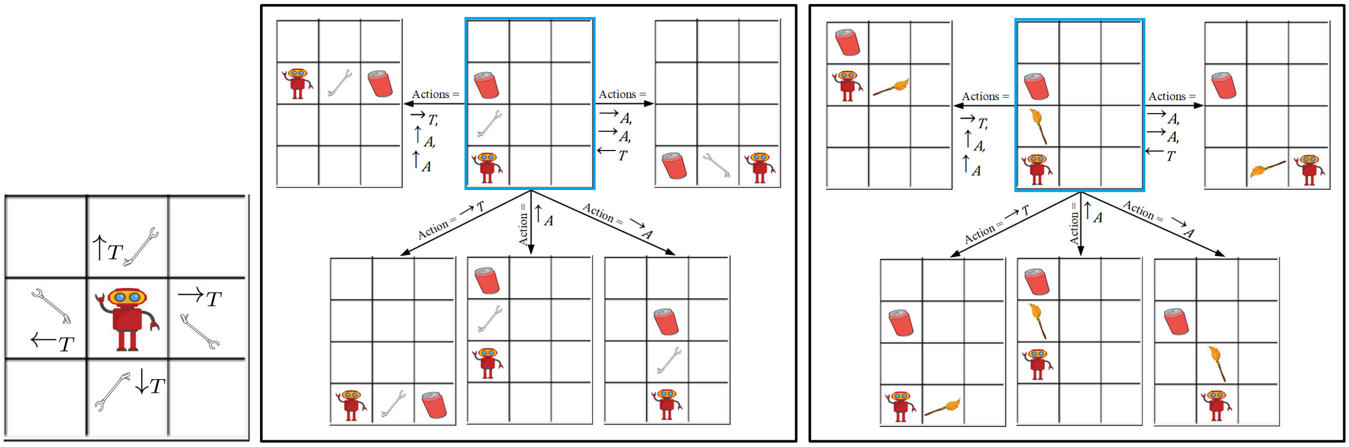
Here, the transitions underlying the grid-worlds' dynamics T are deterministic. Hence, all the subsequent empowerment computations can be carried out using Equation (2). According to T, the agent cannot move itself, the tool, and the object beyond the edges, nor over the walls, of the grid-world. So, if the agent tries to do so, the state will not change. Then, if the agent is next to the can (i.e., $d_M(s^{\mathfrak{A}}, s^{\mathfrak{O}}) = 1$), the agent can use its actions a_A to push it into an adjacent cell. In particular, if the agent is adjacent to the can and the agent moves one step in the direction of the can, the can is pushed one step into the same direction (and remains adjacent to the agent). In doing so, repeatedly moving and pushing the can towards its direction of motion, the agent can move the can around the grid-world. Since the agent can only push the can forward, in case it needs to push the can in a different direction, it needs to spend a few steps to align itself in the right direction, so that that its subsequent push will drive the can towards the new desired route. However, if the agent equips the *picker tool* and needs to change direction of motion while the can is attached to the picker, to bring the can with it by executing a_T costs only one step to the agent, because a_T instantaneously moves the picker in the desired direction (see Fig. 1a). In particular, if the agent, the picker and the can are vertically or horizontally aligned and the equipped tool is adjacent to the can (i.e., $d_M(s^{\mathfrak{T}}, s^{\mathfrak{O}}) = 1$), we will say that the can is "attached" to the picker. This implies that, if the agent moves using a_A, it will carry with it both the picker and the can attached to it. Moreover, if the agents moves the tool by executing a_T, it will rotate both the picker and the can attached to it (see Fig. 1b for some illustrative examples of the picker transitions)[1]. In few words, by picking and carrying the can with the picker tool, the agent can move the can in different locations of the environment quicker than by pushing the can with its own body.

The objective of the two investigated episodic MDPs is to move the can to the waste bin position (i.e., the goal cell) by employing the minimum number of steps, whether with or without the help of a tool. To this aim, the reward function R is defined as follows: when the agent successfully moves the can to the waste bin (i.e., $s^{\mathfrak{O}} = s_g$), it receives an instantaneous reward of 0, otherwise it receives a reward of -1 for every other transition. Notably, moving into a wall does not result in any additional penalty and just incurs the time cost. In order to solve the MDPs considered in our experiments, we employed the deep RL method Advantage Actor Critic (A2C) [21].

B. First Experiment - Fully Observable VS Object Empowerment

In this experiment, we deployed the agent in the environment shown in Fig. 3a, which depicts the initial state of the MDP (we will discuss the colours and values within the cells later). To solve the task, the agent can either go to the can and directly pushing it towards the waste bin, or it can first exit the "corridor" to equip the picker and then use it to bring the can into the waste bin. Although it seems that the agent is wasting steps by going to pick up the tool before approaching the can, the advantages of moving the can with the help of the picker turn the strategy of using the tool into the optimal behaviour, with a minimum number of steps of 15. This is related with the average agent's object empowerment ($\hat{\mathfrak{E}}^6_{\mathfrak{A}\mathfrak{O}} = 0.6$ bits) being smaller

[1] Note that when the tool is equipped, all $a \in \mathcal{A}$ can be considered "tool actions" in the sense expressed in Section III-B.2 (i.e., $\mathcal{A}^{\mathfrak{T}\mathfrak{O}} = \mathcal{A}$ and $\mathcal{A}^{\mathfrak{A}\mathfrak{O}} = \emptyset$), because both agent and tool movements can change the state of the tool $s^{\mathfrak{T}}$ while the can is attached. On the contrary, when the tool is not equipped, none of the action in $\mathcal{A}$ have any impact on the state of tool $s^{\mathfrak{T}}$ and so $\mathcal{A}^{\mathfrak{A}\mathfrak{O}} = \mathcal{A}_A$ and $\mathcal{A}^{\mathfrak{T}\mathfrak{O}} = \emptyset$.



(a) Tool actions.

(b) Five examples of the picker tool's transitions.

(c) Five examples of the broom tool's transitions.

Fig. 1: The tool actions and the transitions of the picker and the broom tools. The grids with blue color borders represent the starting state of an agent.

than the average TOE ($\hat{\mathfrak{E}}^{6}_{\mathfrak{T}\mathfrak{O}} = 1.4$ bits). This means that the picker has an impact on the state of the can (i.e., $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}\mathfrak{O}} > 0$) and that this impact is larger than the one of the agent without the tool. In other words, there are things that the agent can only do by using the picker. Note that the small magnitude of $\hat{\mathfrak{E}}^{6}_{\mathfrak{T}\mathfrak{O}}$ reflects the fact that only when the agent is close enough to the can, it can move the latter by using the picker. In general, tools that can impact the object from a larger distance will have larger $\hat{\mathfrak{E}}^{h}_{\mathfrak{T}\mathfrak{O}}$.

1) Results: We conducted a performance comparison among three RL agents: a standard RL agent, one utilizing FOE as a regulariser (see eq. (1)) and another agent employing TOE as a regulariser. In Table I, we report the average number of episodes needed for each agent to converge towards the optimal solution. As convergence criterion, we verified that the value of the double moving average of the return across episodes was within 0.9 from the optimal return (the average windows' sizes were 100 and 2000). The average was computed across 10 independent runs. The table shows that the agent employing $\mathfrak{E}^{6}_{\mathfrak{T}\mathfrak{O}}$ ($\beta = 0.07$) converged faster than the other agents, with an average number of episodes equalling 28841.4. The second fastest agent was the one regularised by $\mathfrak{E}^{5}_{S}$ ($\beta = 0.09$). Finally, the standard RL agent with no regularisation was clearly the slowest agent.

TABLE I: The average number of episodes when convergence occurred in the first environment.

Approach	Avg. no. of episodes ± std
A2C	37438.0 ± 3461.7
A2C with 5-step FOE	31003.3 ± 3803.2
A2C with 6-step TOE	**28841.4 ± 2420.0**

2) Discussion: We have seen that the agents encouraged by empowerment perform better than the standard RL agent, confirming that regularisation methods based on IM signals can be useful to counteract the sparsity of reward characterising tasks such as the one considered (i.e., here the agent receives a reward different from 0 only when the task is completed). But not all IMs are the same. The results have shown that agents employing FOE and TOE have distinct performances. We believe this difference can be understood by investigating how these IMs shape the agent behaviour in terms of its interaction with the tool and the object present in the environment.

FOE being a measure of the number of future states visited by the agent, the simple option of being able to change the state of the picker, when this is equipped, increases its value. This can be seen in Fig. 3, where we reported three snapshots of empowerment landscapes for fixed tool and object positions. Here, the coloring of the states reflects their empowerment, whose values in bits are reported within the corresponding cell and in the color bars. In Fig. 3a, we can observe that $\mathfrak{E}^{1}_{S}$ is particularly large in states surrounding the tool, with values that range from 2.6 to 3 bits. Hence, adding $\mathfrak{E}^{1}_{S}$ to the reward makes those states a beacon that attracts the agent towards the tool, helping the agent to find it and equip it. In Fig. 3b, we have increased the FOE horizon h to 5. When compared to $\mathfrak{E}^{1}_{S}$, we can see that a larger h implies higher values of FOE, with $\mathfrak{E}^{5}_{S}$ having a maximum of 7 bits near the picker. Furthermore, in Fig. 3b the larger values of $\mathfrak{E}^{5}_{S}$ are equally distributed between the states surrounding both the picker and the can, because in this area 5 steps are enough to both equip the tool and move it, or to push the can around with the agent's actions $\mathcal{A}_A$. Hence, to maximise $\mathfrak{E}^{5}_{S}$ can also encourage the agent to move towards the can.

Object empowerment measures the number of positions the agent is able to move the can to. Hence, we observed that in the grid-world the states with the largest TOE $\mathfrak{E}^{1}_{\mathfrak{T}\mathfrak{O}}$ of 1 bit are those located in the cells next to the can, whereas all other states have $\mathfrak{E}^{1}_{\mathfrak{T}\mathfrak{O}}$ equal to 0 bits. However, if the agent would be only influenced by $\mathfrak{E}^{1}_{\mathfrak{T}\mathfrak{O}}$, it may go to the can before picking the tool without executing the optimal solution. Interestingly, if the TOE's horizon h is large

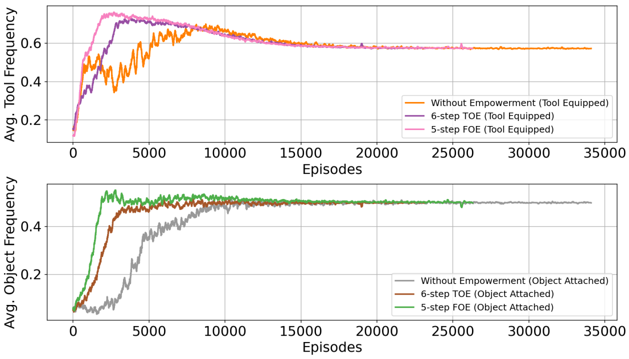
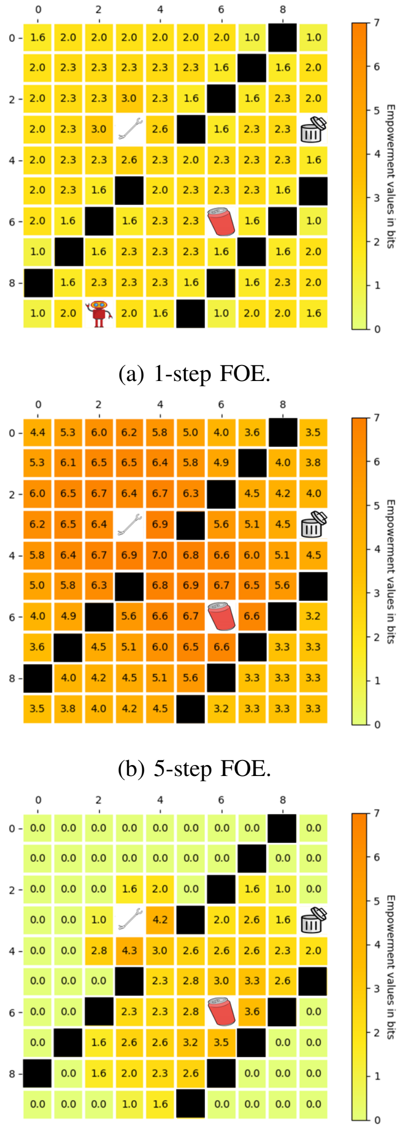


Fig. 2: The average proportion of time steps at which the agent has the picker equipped (fuchsia, purple, and orange curves) and at which the can is attached to the picker (green, brown, and grey curves) in the first environment.

enough, the action sequences starting in the states next to the picker will include the actions that bring the agent with the tool to the can, together with the subsequent actions that allow the agent to move the can with the picker (see Fig. 3c). Since the picker enables a large influence on the can's state, in this case $\mathfrak{E}^{h}_{\mathfrak{T}\mathfrak{O}}$ can be higher next to the picker than it is next to the can. This phenomenon is visible in Fig. 3c, where the maximum value of $\mathfrak{E}^{6}_{\mathfrak{T}\mathfrak{O}}$ next to the picker is 4.3 bits, while it is 3.6 bits next to the can. We observed that not only TOE with a large enough h can steer the agent to the tool but, once this is equipped, it can subsequently encourage the agent to move towards the can, because TOE will be larger as the agent approach it.

To confirm that the usage of FOE and TOE as intrinsic rewards encourages the agent to approach the picker, in Fig. 2 we show the proportion of time during which the picker is equipped by the agent, averaged across 10 runs for each episode. The results show that at the initial stage of learning, when compared with a standard RL agent with no regularisation (orange curve), the agents that employ $\mathfrak{E}^{5}_{S}$ and $\mathfrak{E}^{6}_{\mathfrak{T}\mathfrak{O}}$ as intrinsic rewards (fuchsia and purple curves, respectively) spend more time on average with the tool equipped. Since once the tool is picked this remains equipped until the end of the episode, the plots indicate that the intrinsically motivated agents find the tool earlier than the standard agent. Furthermore, the agent regularised by $\mathfrak{E}^{5}_{S}$ equips the tool sooner that the agent regularised by $\mathfrak{E}^{6}_{\mathfrak{T}\mathfrak{O}}$. In Fig. 2 we also report the average proportion of time, after the tool is equipped, spent by the intrinsically motivated agents while the can is attached to the picker. The figure shows that the agents employing $\mathfrak{E}^{5}_{S}$ (green curve) and $\mathfrak{E}^{6}_{\mathfrak{T}\mathfrak{O}}$ (brown curve) spend more time on average with the can attached to the picker than the vanilla agent (grey curve). Since once the can is attached to the picker this remains attach until the end of the episode, the latter finding shows that the intrinsically motivated agents attach the can before than the vanilla agent. Moreover, the agent motivated by $\mathfrak{E}^{5}_{S}$ picks the can before the agent motivated by $\mathfrak{E}^{6}_{\mathfrak{T}\mathfrak{O}}$. We believe that the combined attractivity of both the tool and the object helps to interpret the good performance of the agents regularised by $\mathfrak{E}^{5}_{S}$ and $\mathfrak{E}^{6}_{\mathfrak{T}\mathfrak{O}}$. We also think that the TOE agent has a better RL performance than the FOE agent (Table I) because the latter may struggle in bringing the can to the goal cell, because this is located in a state with low $\mathfrak{E}^{5}_{S}$ [22].



(a) 1-step FOE.

(b) 5-step FOE.

(c) 6-step TOE.

Fig. 3: 1- and 5-step FOE, and 6-step TOE in the first environment.

C. Second Experiment - Tools Comparison

For this second scenario, we placed the agent within the environment illustrated in Fig. 4a, which shows the starting state of the task. Here, we are interested in using empowerment to compare the impact that two different tools have on the state of the object, and in how this influence is reflected into the performance of agents equipping these tools. To this aim, in addition to the picker tool, here we introduce the *broom tool*. Consider that only one of the two tools is used within the same simulation, and that this is already equipped by the agent at the beginning of every episode (i.e., here tools do not need to be picked up). While the broom can be moved by the agent in the same way the picker does by using the actions a_T, the two tools differ in the way they interact with the can. When coming in contact with the can, the picker acts "sticky" and will henceforth move the object with it, while when pushing the can the broom merely extends the effective body of the agent and can expand the latter in the desired direction, but does not stick with the can. Hence, when the broom and the can are one next to each other, the can does not get attached to the tool, so if the agent moves away the can remains in its cell. On the contrary, if the agent with the broom pushes in the direction of the can, the latter will slide of one cell along the direction of motion (refer to Fig. 1c to view a few illustrative examples about the broom transitions). This tool does not offer to the agent additional ways of moving the can, if not by its ability of pushing the can sideways when this is located diagonally from the agent's location (something that the agent cannot do). This statement can be formalised by showing that the average broom's TOE (denoted by $\hat{\mathfrak{E}}^3_{\mathfrak{T}\mathfrak{B}\mathfrak{O}}$ and equals to 0.28 bits) is slightly larger than the average agent's object empowerment ($\hat{\mathfrak{E}}^3_{\mathfrak{A}\mathfrak{O}} = 0.23$ bits). Furthermore, the average TOE can be used to assess that the picker (represented by $\hat{\mathfrak{E}}^3_{\mathfrak{T}\mathfrak{P}\mathfrak{O}}$ and equals to 0.52 bits) has a larger impact on the can's state than the broom and than the agent on its own.

1) Results: We performed a comparative analysis of four RL agents: a vanilla agent and an TOE-regularised agent, both equipping the picker; a vanilla agent and an TOE-regularised agent, both equipping the broom. The optimal return obtained by moving the can in the waste bin with the picker is -10 and with the broom is -12. Table II demonstrates how the broom is an "inefficient" tool, because the agent with the picker utilizing $\mathfrak{E}^3_{\mathfrak{T}\mathfrak{P}\mathfrak{O}}$ achieved a faster convergence (15555 episodes with $\beta = 0.08$) than the agent with the broom utilizing $\mathfrak{E}^3_{\mathfrak{T}\mathfrak{B}\mathfrak{O}}$ (37658 episodes with $\beta = 0.08$). Furthermore, also in the case of the standard RL agents without any regularisation, we observed that with the use of the picker the agent learns quicker than with the use of the broom.

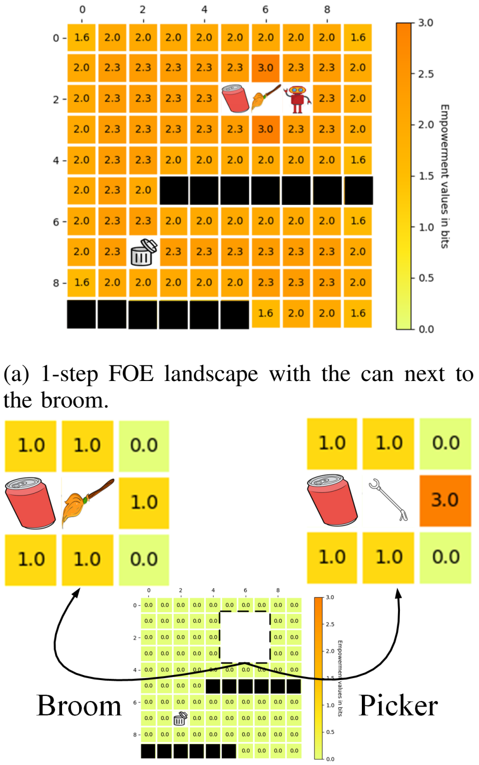
TABLE II: The average number of episodes when convergence occurred in the second environment.

Approach	Avg. no. of episodes ± std
A2C (Picker)	15555.4 ± 418.8
A2C with 3-step TOE (Picker)	**11672.4 ± 733.4**
A2C (Broom)	52872.2 ± 7742.2
A2C with 3-step TOE (Broom)	**37657.6 ± 4885.9**

2) Discussion: In Table II, we can observe that the intrinsically motivated agent with the fastest convergence was the one with the picker, which, between the two tools, it is also the one with the largest average TOE. On the contrary, the regularised agent with the broom took a very long time to converge and, when compared with the picker, its average TOE was much smaller (0.52 bits vs. 0.28 bits). A more "local" view of this can be seen in the TOE values around the picker and the can reported in Fig. 4b. Usually, to use a tool with a larger value of $\hat{\mathfrak{E}}^h_{\mathfrak{T}\mathfrak{O}}$ implies also a larger attraction towards the tool and object positions, which provides an additional boost to the benefits of the empowerment-based regularisation.

In principle, both FOE and TOE could be used to compare the picker and broom tools, but FOE has some undesirable properties. For instance, let us consider the $\mathfrak{E}^1_S$ landscape of Fig. 4a, where the broom is next to the can. If we would have replaced the broom with the picker, we would have obtained exactly the same landscape. This happens because FOE is not able to discriminate the number of possible tool states (which is the same for the broom and the picker) from the number of possible tool states that imply also a change in the can's state (which is lower for the broom than for the picker). So, when h is low, $\mathfrak{E}^h_S$ is not able to distinguish between the broom and the picker (the same holds for the average FOE $\hat{\mathfrak{E}}^1_S$), while these tools have different impact on the object. In Fig. 4b, we can see that this is not the case with $\mathfrak{E}^1_{\mathfrak{T}\mathfrak{O}}$, which has maximum value next to the picker equal to 3 bits and next to the broom equal to 1 bit. Furthermore, if we move the can far away from the broom, the values of 1-step FOE around the broom are identical to those of the broom being next to the can (Fig. 4a). So, when h is low, $\mathfrak{E}^h_S$ is not able to distinguish between the tool being near or far from the object. As opposed to that, when h is low and the can is far from the tool, $\mathfrak{E}^h_{\mathfrak{T}\mathfrak{O}}$ will be 0 bits in the states surrounding the tool and different from 0 only in the states surrounding the can. For larger h, the $\mathfrak{E}^h_S$ of states near the tools, and the corresponding average $\hat{\mathfrak{E}}^h_S$, can be even larger for the broom than for the picker. This happens because with the broom the agent can place the can in cells detached from the tool, while in the case of the picker the tool and the can are always attached until the end of the episode. On the contrary, we observed the average values of $\hat{\mathfrak{E}}^h_{\mathfrak{T}\mathfrak{O}}$ for the picker are always larger than the ones of the broom, and their discrepancy grows as h becomes higher.

In general, the relationship between the $\hat{\mathfrak{E}}^h_{\mathfrak{T}\mathfrak{O}}$ of an agent using a certain tool to manipulate an object, and the agent's performance in problems involving the object, can be task-dependent. Certain tasks involving a given object might not benefit at all from tools with large TOE. In fact, the states of the object needed to solve the task may not be part of the future states measured by empowerment. However, from a skill learning perspective [13], it is often desirable to master the interaction with an object as much as possible, even before knowing the exact nature of the task that involves the object. More diverse configurations of an object are covered by the agent's pre-trained skills, such as using a tool with large $\hat{\mathfrak{E}}^h_{\mathfrak{T}\mathfrak{O}}$, and more it will be probable that one of them will be useful to solve, if not all, many of the downstream tasks involving the object. In this regard, we believe that TOE is a good measure to compare different tools with respect to their potential applications on a specific object of the environment,



	0		2		4		6		8	
0	1.6	2.0	2.0	2.0	2.0	2.0	2.0	2.0	2.0	1.6
	2.0	2.3	2.3	2.3	2.3	2.3	3.0	2.3	2.3	2.0
2	2.0	2.3	2.3	2.3	2.3				2.3	2.0
	2.0	2.3	2.3	2.3	2.3	2.3	3.0	2.3	2.3	2.0
4	2.0	2.3	2.3	2.0	2.0	2.0	2.0	2.0	2.0	1.6
	2.0	2.3	2.0							
6	2.0	2.3	2.3	2.0	2.0	2.0	2.0	2.0	2.0	1.6
	2.0	2.3		2.3	2.3	2.3	2.3	2.3	2.3	2.0
8	1.6	2.0	2.0	2.0	2.0	2.0	2.3	2.3	2.3	2.0
							1.6	2.0	2.0	1.6

(a) 1-step FOE landscape with the can next to the broom.

(b) 1-step TOE landscape with the broom and the picker next to the can.

Fig. 4: landscapes of second environment.

because it quantifies the impact a certain tool has on the state of the object.

V. Conclusions

In this study, we have shown that the intrinsic motivation empowerment is particularly suitable for tasks that involve tools. We applied this framework to simple grid-world environments, where an agent had to interact with objects using different tools. In particular, a key concept in our study is *object empowerment*. This is a measure of how much the agent can influence an object. We found that there are certain things an agent can do only by using a tool, which we formalised using the concept of average object empowerment. This means that the agent's ability to influence an object can be greater when it uses a tool, compared to when it doesn't use a tool. Furthermore, object empowerment can be used to identify which tool is better for interacting with the object, meaning which tool can influence the object the most. We also employed object empowerment as an intrinsically motivated regulariser to guide the RL agent towards states beneficial for learning how to master tools for efficient task completion. Our results suggest that this is a promising first step towards a proper understanding of how agents can use tools and interact with objects in their environment. In the future, we would like to explore scenarios with many objects and many tools. In such a scenario, we could consider all possible pairs of tools and objects, and use the TOE to determine which tool is most effective for each object. Finally, we look forward to potential applications of our framework in the field of robotics for tasks that involve the use of tools. We believe that our work lays a solid foundation for these future investigations.

References

[1] A. H. Taylor and R. D. Gray. Is there a link between the crafting of tools and the evolution of cognition? Wiley Interdisciplinary Reviews: Cognitive Science, vol. 5, no. 6, pp. 693–703, December 2014.

[2] J.H. Flavell, Cognitive development: Children's knowledge about the mind, Annual review of psychology, vol. 50, no. 1, pp. 21-45, February 1999.

[3] A.S. Klyubin, D. Polani and C.L. Nehaniv, Empowerment: A universal agent-centric measure of control, in 2005 IEEE congress on evolutionary computation, vol. 1, pp. 128-135.

[4] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Wiele, V. Mnih, N. Heess and J.T. Springenberg, Learning by playing solving sparse reward tasks from scratch, in 2018 Int. Con. on Machine Learning (ICML), pp. 4344-4353.

[5] S. Mohamed, & D.J. Rezende, Variational information maximisation for intrinsically motivated reinforcement learning, in 2015 Advances in Neural Information Processing Systems (NeurIPS).

[6] J. Sinapov and A. Stoytchev, Learning and generalization of behavior-grounded tool affordances, in 2007 IEEE 6th Int. Conf. on Development and Learning (ICDL), pp. 19-24.

[7] R. Jain and T. Inamura, Learning of tool affordances for autonomous tool manipulation, in 2011 IEEE/SICE Int. Symp. on System Integration (SII), pp. 814-819.

[8] A. Gonçalves, J. Abrantes, G. Saponaro, L. Jamone and A. Bernardino, Learning intermediate object affordances: Towards the development of a tool concept, in 2014 Joint IEEE 4th Int. Conf. on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), pp. 482-488.

[9] K. Seepanomwan, V.G. Santucci and G. Baldassarre, Intrinsically motivated discovered outcomes boost user's goals achievement in a humanoid robot, in 2017 Joint IEEE Int. Conf. on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), pp. 178-183.

[10] K. Seepanomwan, D. Caligiore, K.J. O'Regan and G. Baldassarre, Intrinsic motivations and planning to explain tool-use development: A study with a simulated robot model, IEEE Trans. on Cognitive and Developmental Systems, vol 14, no. 1, pp. 75-89, May 2020.

[11] S. Forestier, R. Portelas, Y. Mollard and P.Y. Oudeyer, Intrinsically motivated goal exploration processes with automatic curriculum learning, Journal of Machine Learning Research, vol. 23, no. 1, pp. 6818-6858, January 2022.

[12] C. Salge, C. Glackin and D. Polani, Empowerment–an introduction. Guided Self-Organization: Inception, 2014, pp. 67-114.

[13] B. Eysenbach, A. Gupta, J. Ibarz and S. Levine, Diversity is all you need: Learning skills without a reward function, arXiv preprint arXiv:1802.06070, 2018.

[14] R. Zhao, Y. Gao, P. Abbeel, V. Tresp and W. Xu, Mutual information state intrinsic control, in 2021 9th Int. Conf. on Learning Representations (ICLR).

[15] S. Dai, W. Xu, A. Hofmann and B. Williams, An empowerment-based solution to robotic manipulation tasks with sparse rewards, Auton. Robots, vol. 47, no. 1, pp. 1-17, February 2023.

[16] R.S. Sutton and A.G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[17] G. Baldassarre, What are intrinsic motivations? A biological perspective, in 2011 IEEE Int. Conf. on Development and Learning (ICDL), pp. 1-8.

[18] G. Baldassarre, T. Stafford, M. Mirolli, P. Redgrave, R.M. Ryan and A. Barto, Intrinsic motivations and open-ended development in animals, humans, and robots: an overview, Frontiers in Psychology, vol. 5, pp. 985, September 2014.

[19] A. Aubret, L. Matignon, and S. Hassas, An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey, Entropy, vol. 25, pp. 327, February 2023.

[20] T.M. Cover and J.A. Thomas, Elements of information theory, John Wiley & Sons, 2006.

[21] L. Graesser and W.L. Keng, Foundations of deep reinforcement learning. Addison-Wesley Professional, 2019.

[22] N. C. Volpi and D. Polani, Goal-directed empowerment: combining intrinsic motivation and task-oriented behaviour, IEEE Trans. on Cognitive and Developmental Systems, vol. 15, no. 2, pp. 361-372, December 2020.