EValueAction: a proposal for policy evaluation in simulation to support interactive imitation learning

EValueAction: a proposal for policy evaluation in simulation to support interactive imitation learning

Fiorella Sibona*, Jelle Luijkx†, Bas van der Heijden†, Laura Ferranti†, Marina Indri*
* Dipartimento di Elettronica e Telecomunicazioni (DET) - Politecnico di Torino, Italy
{fiorella.sibona, marina.indri}@polito.it
† Cognitive Robotics department (CoR) - Delft University of Technology, The Netherlands
{j.d.luijkx, d.s.vanderheijden, l.ferranti}@tudelft.nl

***Abstract*—The up-and-coming concept of Industry 5.0 foresees human-centric flexible production lines, where collaborative robots support human workforce. In order to allow a seamless collaboration between intelligent robots and human workers, designing solutions for non-expert users is crucial. Learning from demonstration emerged as the enabling approach to address such a problem. However, more focus should be put on finding safe solutions which optimize the cost associated with the demonstrations collection process. This paper introduces a preliminary outline of a system, namely EValueAction (EVA), designed to assist the human in the process of collecting interactive demonstrations taking advantage of simulation to safely avoid failures. A policy is pre-trained with human-demonstrations and, where needed, new informative data are interactively gathered and aggregated to iteratively improve the initial policy. A trial case study further reinforces the relevance of the work by demonstrating the crucial role of informative demonstrations for generalization.**

***Index Terms*—Human-centered manufacturing, Learning from Demonstration, Interactive imitation learning, Simulation**

I. Introduction and state of the art

In recent years, the *Industry 4.0* paradigm introduced some impactful technologies for the industrial workflow. Among these, Artificial Intelligence (AI) at large laid the foundations for implementing intelligent autonomous agents. Indeed, machine and deep learning algorithms coupled with smart sensors led to advanced human-robot perception, enabling new ways to attain safe and effective collaboration [1]. Also, simulation and virtual representations of physical systems are becoming a valuable tool to get insights on assets behaviour and to enable powerful industrial digital twins [2], [3].

Moreover, collaborative robots can help humans streamline the manufacturing process. However, human complex and creative reasoning is yet to be achieved by machines. Thereby, current and fore-coming research is investigating a human-centric vision of production lines, the so-called *Industry 5.0*, where more sustainable and value-driven production processes are created giving greater relevance to human skills [4]. When considering human-robot collaborative applications, the cognitive mismatch between a collaborative robot (*cobot*) and a human worker can be narrowed using AI, enabling the robot to interpret and adapt to the worker's behaviour. Nevertheless, the user is typically required to have the expertise necessary to understand and change the robot's behaviour. This limitation can be overcome by exploiting Learning from Demonstration (LfD), or *imitation learning* (IL), where a human teacher demonstrates to the robot how the task should be executed.

Among the issues affecting LfD we have: (i) *dataset bias*, linked to the teacher-specific behaviour or the lack of variety of demonstrated situations and, (ii) *overfitting*, caused by too fine-tuned models or data scarcity in terms of the number of provided demonstrations. These issues hamper the generalization, or *extrapolation*, capability of LfD algorithms, resulting in undesired behaviour whenever new situations are met. To avoid such problems, the human would be required to (i) know how to offer informative and unbiased demonstrations, provided in heterogeneous set of situations, and (ii) identify and provide a sufficient number of demonstrations. This results in additional mental and physical effort on the human teacher side. The time that the human spends on performing demonstrations can be foreseen to be included in the future non-value adding activities, i.e., activities hindering the transition towards lean manufacturing paradigms [5]. Also, the authors of [6] point out that robot learning metrics should focus on time efficiency, to better reflect the *true cost for humans*.

Therefore, achieving generalization with few informative demonstrations is one of the main drivers of research in the field of LfD. What emerged from the analysed literature is that, independently of the inherent generalization capability of the proposed methods, the quantity and quality of demonstrations greatly impact the achievable extrapolation. It is then possible to derive two main objectives to reduce the *cost of generalization* (from the human point of view): reduce the number and improve the quality of demonstrations.

The following works aim at improving generalization while keeping a low number of demonstrations. Some works try to achieve adaptability by incorporating into the model a set of task variables describing the context under which demonstrations were performed [7], or conditioning the learning process on different information sources conveying the task [8]. Other works exploit dataset augmentation to achieve policy improvements for learning task-parameterized skills without increasing the number of demonstrations, such as [9], in which noise is added on recorded paths, and [10] where generated synthetic data are added to the dataset. Several other works, as [11] and [12], take advantage of reinforcement learning (RL) to explore and adapt the model to new situations, after a first phase learning from few demonstrations. In [13], goal proximity is used as a dense reward for the agent training.

Moreover, human demonstrations quality can affect the ability to achieve good extrapolation [14]. In fact, obtaining high-quality demonstrations and providing a definition for

The authors would like to thank Anna Mészáros and Giovanni Franzese, both with the CoR department at Delft University of Technology, for the invaluable support provided during algorithm adaptation to the case study. This work was supported by the European Union's H2020 project Open Deep Learning Toolkit for Robotics (OpenDR) under grant agreement #87144.

such quality appear among the most critical challenges when learning from offline human data [15], [16]. Nevertheless, some works try to exploit all available demonstrations, independently of their quality, to exploit a larger number of data. In [17], the proposed framework learns a well-performing policy also including confidence-reweighted non-optimal demonstrations. The method proposed in [18] computes a generalized trajectory from all demonstrations, while the authors of [19] fully exploit the demonstrations dataset to build a compositional task latent representation space.

The similarity-aware framework presented in [20], evaluates different representations based on a defined similarity, and then provides the user with the most similar reproduction of the demonstrated skill for unseen conditions. Also, the authors of [21] aim at extending the extrapolation capabilities of IL methods by means of virtual demonstrations, generated with the invariants method, to represent a better consistency and quality alternative to human demonstrations.

The quality and quantity of demonstration goals for good generalizing policies can be attained by a system seeking to guide the user's demonstrations interactively. In Interactive Imitation Learning (IIL) human demonstrations are periodically provided during robot execution. Compared to standard IL training, IIL can be more sample-efficient since demonstrations, in this context known as *feedbacks*, *corrections*, or *interventions*, are also collected while executing the novice policy, rather than the teacher's policy alone [22]. The method in [23] exploits the epistemic and aleatoric uncertainty information to detect ambiguities in the feedbacks to support the process of finding the best learning samples. The algorithm in [24] exploits topological persistence to detect ambiguity in a trained policy, in order to query user demonstrations only if needed, avoiding to gather demonstrations for known situations. A family of robot-gated IIL methods, stemming from DAgger [25], allows the agent to query a teacher intervention according to estimations of quantities related to task performance and uncertainty [26], [27]. In particular, ThriftyDAgger [28], considers the state novelty and the state risk of task failure to trigger corrections, given a budget of human interventions.

The EValueAction (EVA) framework, whose concept idea is introduced in this paper, has been designed to represent a support for the interactive learning process by guiding the user towards the most informative demonstrations. A state risk of failure is inferred by computing in simulation an estimate of its value function, given the executed policy. This way, new demonstrations are queried where needed, and failures are foreseen and prevented before acting in the real world, improving safety. Also, the case study thoroughly analysed in this paper provides a clear display of the dependence of generalization on the teacher expertise and resulting demonstrations' execution. The remainder of this paper is organized as follows: Section II describes the case study that brought to the research idea. Then Section III outlines the concept EVA system, followed by a possible solution for policy evaluation in simulation. Finally, Section IV draws some conclusions and sketches future research directions.

II. Preliminaries

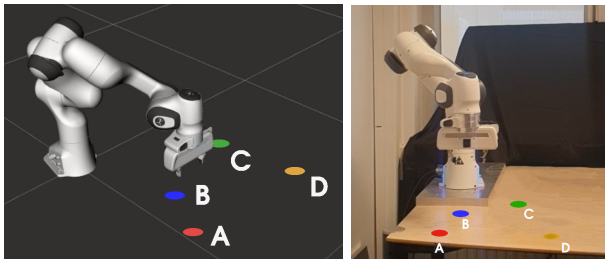
Before outlining the proposed EVA system, it is worth describing the case study that brought to the concept idea. Given the following problem contextualization, the assumptions and trials outcomes are reported. In the context of flexible manufacturing, the collaborative assembly task represents a manufacturing operation of high practical relevance: the robot is typically allocated repetitive or power demanding tasks, while the human performs highly dexterous operations [29]. In such tasks, being able to adapt to new situations is crucial, as the workspace configuration and the positions and shapes of assembly parts often vary. Following the assembly task subdivision introduced in [30], we consider an *approaching phase* and an *assembling phase*. The authors perform such division in order to prevent the potential under-fitting caused by the variability of the demonstration data between the two assembly stages. In particular, the approaching phase is the one most affected by environment constraints and configuration of parts. As such, this stage would take the most advantage from an IL algorithm with good generalization capabilities learning from few demonstrations. Therefore, we consider a human teaching a cobot how to perform a desired approaching phase. For example, we can imagine a peg-in-hole collaborative assembly task, where the approaching phase can be reduced to a pick-and-place or hovering task.

As extensively reported, improving generalization is widely tackled by researchers, as poor generalization capabilities is a common issue in IL. The case study described hereafter provides a clear idea of how generalization capabilities of IL methods can be greatly affected by the way demonstrations are performed.

1) A simplified learning scenario: For the execution of the trials, we have taken advantage of the algorithm presented in [31]. Specifically, we exploited the Interactive Learning of Stiffness and Attractors (ILoSA) framework to learn attractors only from kinesthetic demonstration, without taking advantage of the interactive part. In ILoSA, the confidence level provided by the use of Gaussian Processes (GPs) allows to detect when the cobot lands on an unknown state, and is exploited to implement a stabilizing attractive field to lead the robot towards minimum variance regions. As ILoSA recorded demonstrations within a global frame, generalization was limited around the visited states. Hence, we borrowed the general idea of using local reference frames, as hinted in [32], to be able to test over different final positions.

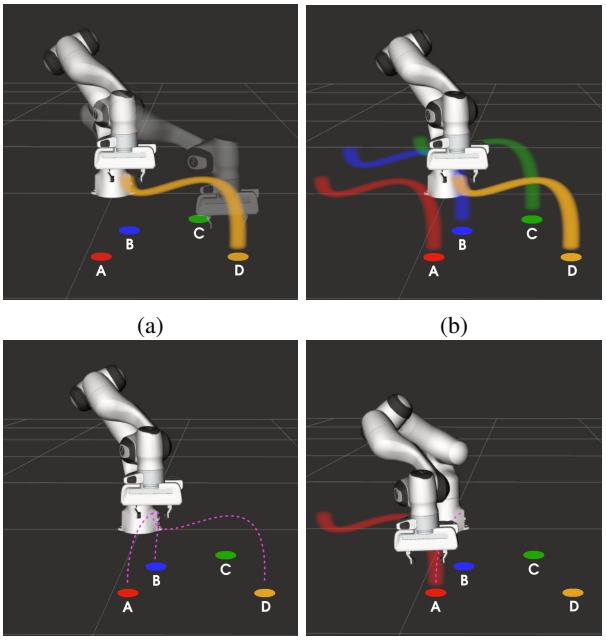
In summary, we exploited ILoSA to learn from human demonstrations the attractor distance for the robot impedance control, learned with respect to the final position reference frame. The focus of the trials was on reaching as many new goals as possible with one single informative demonstration.

2) Demonstrations setup: The demonstrations and trials have been performed on a 7 DOF Franka-Emika Panda with an impedance controller and a ROS communication network. The demonstration consisted in moving the robot from a reference home position to a final position of interest. For our approaching phase scenario, we assume that an assembly part is picked from some location and brought to the reference home position at each assembly task iteration. Then the final desired position (before assembly stage) is reached exploiting the learned policy. As expected, the employed IL algorithm was able to generalize in the neighbourhood of a demonstrated trajectory task. Therefore, to check the generalization behaviour over



(a) Points of interest in *rviz*. (b) Real setup.

Fig. 1: Four goals of interest on the working plane have been chosen for the generalization check trials.



(a) (b)

(c) (d)

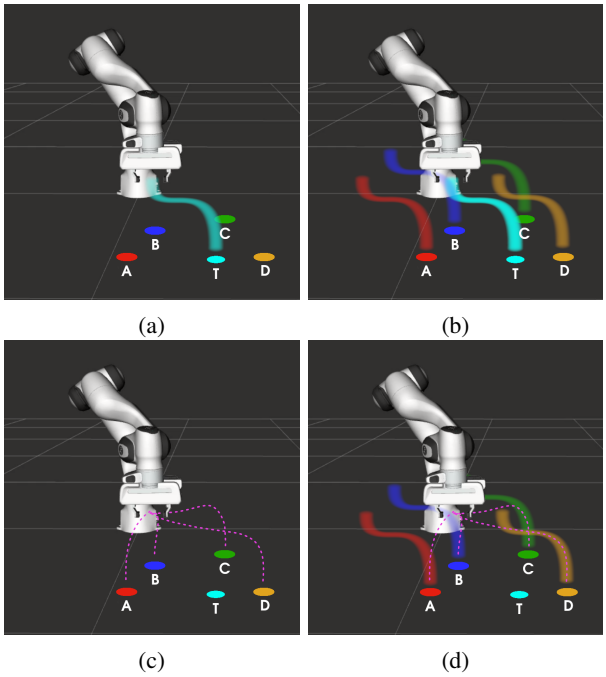
Fig. 2: Medium expertise demonstration on `D`.

destination points relatively far from the demonstrated one, we have chosen four goals of interest (Figure 1).

a) Automatic demonstration recording: The ILoSA Jupyter interactive Python code has been modified so as to interactively input a final goal point to accordingly name saved data and trained GPs models files/structures, while automatically generating a `.txt` file to keep track of the tests and relative outcomes. Also, after recording, the produced code lets iteratively test the learned policy over the set of interest points. This procedure allowed for faster data collection and consultation.

3) Trials outcomes: The execution of several trials allowed to identify three relevant cases that we describe hereafter. Assumption: after trials, demonstrations with points `A` and `B` as final destination turned out to be the least generalizing thus are not considered as demonstration points.

a) Medium expertise demonstration: Destination hovering point is one among `A`, `B`, `C`, and `D` (refer to Figure 2). A human teacher performed a demonstration with the position above point `D` as the final position to be reached (Figure 2a). Note that the height recorded during the single demonstration



(a) (b)

(c) (d)

Fig. 3: Expert demonstration on `T`.

was kept as a hovering height for all worktable points set as desired goals. After the policy training, the motion was learned in the local reference frame leading to areas with high confidence for each final goal, as shown in Figure 2b. When the policy was tested on all the points of interest, it generalized well on `A` and `B`, since the IL algorithm attracted the robot towards the high confidence areas thanks to its stabilization prior (Figure 2c). Then, when `A` was the input for policy testing, the cobot was able to reach it (Figure 2d).

b) Expert knowledge demonstration: Destination hovering point is not among the points of interest (Figure 3). An expert human teacher, well aware of the IL algorithm behind the training process, demonstrated a motion to reach the position above an empirically chosen test point `T` (Figure 3a), which intuitively would have allowed the trained policy to generalize on all points of interest. Indeed, after GPs training, the policy generalized on `A`, `B`, `C` and `D` (Figure 3b–3d). Note that, in this case, the generalization capability is influenced by the specific motion shape and the stabilization fields, since, depending on both, the robot may or may not be attracted to the locally learned motion.

c) Little knowledge demonstration: We then let a human teacher kept unaware of the underlying IL algorithm demonstrate a motion towards `D`. As the human teacher did not have any information on the influence the motion shape would have had on the generalization capability, the resulting policy surprisingly didn't generalize on the other points of interest, learning the motion to hover on `D` only. Note that the human teacher had expertise in robotics but no information on the learning process.

4) Main takeaways: The performed trials then brought out that with an expert teacher, able to infer the most informative demonstrations with the aim of generalization, a single demonstration could be sufficient to generalize over four points

of interest and their neighbourhoods (plus the testing point for demonstration and relative neighbourhoods). Conversely, if the user is non-expert, with one demonstration only one task would be learned. Independently of the used LfD method and the simplicity of the learning scenario, this gave some intuitions on how robots can help to improve demonstration quality and reduce demonstration quantity. Namely, as the machine is well aware of the policy, it can give some suggestions on where new demonstrations would be most informative and potentially perform virtual demonstrations, thereby reducing the number of demonstrations demanded to the human. This led to the idea of exploiting simulation for policy evaluation, which in turn would also improve safety during collaboration.

III. The EValueAction (EVA) System

The proposed system, EVA, seeks to provide a framework to guide the human teacher during the interactive demonstrations to improve the generalization over new states of a learned policy. The main elements of the system would be: (i) the LfD method of choice, (ii) automatic recording of demonstrations to populate a dataset, (iii) a policy evaluation algorithm exploiting a digital twin, (iv) human-robot interface for bidirectional information exchange.

The overall goal is to let the robot successfully perform a task by taking actions $\boldsymbol{a} \in \mathcal{A}$, given observations $\boldsymbol{o} \in \mathcal{O}$, where $\mathcal{A}$ and $\mathcal{O}$ are the robot's action and observation spaces, respectively. These actions could, for example, be reference end-effector configurations, while observations could comprise joint angle measurements and camera images. These observations result from states $\boldsymbol{s} \in \mathcal{S}$, i.e., $\boldsymbol{o} = O(\boldsymbol{s})$ where $\mathcal{S}$ is the state space and O the observation mapping $O : \mathcal{S} \to \mathcal{O}$. We make this distinction between states and observations, since full state information is often not available in real world scenarios. Since actions follow from observations, the aim is to find a policy π (which is a mapping from observations to actions, i.e., $\pi : \mathcal{O} \to \mathcal{A}$) such that a task is performed successfully. Furthermore, we would like this policy to be successful starting from a set of initial states $\boldsymbol{s}_0 \in \mathcal{S}_0 \subset \mathcal{S}$. That is to say, the robot policy does not have to be successful starting from all possible states, but should be performing well for the actual distribution over initial states it encounters, which we denote by $p(\boldsymbol{s}_0)$. Furthermore, successful completion of a task can be determined by defining a goal state set $\mathcal{S}_g \subset \mathcal{S}$. Note that goals could be dynamic and part of the state $\boldsymbol{s}$. We can quantify the optimality of a policy by defining a reward function $R(\boldsymbol{s})$. Given the reward function, we can define the value of a state [33]:

$$V_\pi(\boldsymbol{s}) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R(\boldsymbol{s}_{t+k+1}) \middle| \boldsymbol{s}_t = \boldsymbol{s} \right], \tag{1}$$

where $\mathbb{E}_\pi[\cdot]$ is the expected value given that policy π is executed, t is any time step and γ is the discount rate with $0 \leq \gamma \leq 1$. Intuitively, the value of a state says how well the policy performs on average starting from that state following policy π. We define the optimal policy π^* as the one that has the highest expected value given our distribution of initial states:

$$\pi^* = \arg\max_{\pi \in \Pi} \mathbb{E}_{\boldsymbol{s}_0 \sim p(\boldsymbol{s}_0)}[V_\pi(\boldsymbol{s}_0)], \tag{2}$$

where Π is the policy space and $\mathbb{E}_{\boldsymbol{s}_0 \sim p(\boldsymbol{s}_0)}[\cdot]$ denotes the expectation given that $\boldsymbol{s}_0$ is drawn from distribution $p(\boldsymbol{s}_0)$. However, in practice it is often not trivial to come up with an appropriate reward function $R(\boldsymbol{s})$. This would particularly be the case if we were to solve this problem with RL. In that case, the reward function should ideally guide the robot towards successful behaviour [34], which would require *reward shaping* [35]. This can be a time consuming process that could also lead to suboptimal behaviour. Robotic RL also results in challenges related to data efficiency and safety [34], [36]. Therefore, we choose to find π^* by LfD. In this setting, it is not required that $R(\boldsymbol{s})$ guides the agent and we can only reward success:

$$R(\boldsymbol{s}) = \begin{cases} 1 & \text{if} \quad \boldsymbol{s} \in \mathcal{S}_g \\ 0 & \text{else} \end{cases}. \tag{3}$$

Worth noting is that with this reward function and $\gamma = 1$, the optimal policy π^* will simply be the policy with the highest success rate. By setting $0 < \gamma < 1$, not only success will be rewarded, but the robot learner will also be stimulated to solve the task in minimum-time. This is desirable in our scenario, as time is related to cost in most industrial applications.

Our method falls within the realm of IIL, hence demonstrations are also interactively gathered while executing the learner policy. This results in demonstrations for states that the robot learner actually encounters, rather than only demonstrations for states the teacher encounters. We can collect a dataset with N demonstration trajectories $\mathcal{D} = \{\boldsymbol{\tau}_0, \boldsymbol{\tau}_1, \ldots, \boldsymbol{\tau}_N\}$ and try to imitate the expert policy. We consider demonstrations that consist of observation vectors, i.e., $\boldsymbol{\tau} = [\boldsymbol{o}_0, \boldsymbol{o}_1, \ldots, \boldsymbol{o}_T]$ where T is the last time step of demonstration $\boldsymbol{\tau}$. An estimate of the optimal policy can be obtained by minimizing a loss between the demonstrated trajectories and trajectories resulting from the policy:

$$\hat{\pi}^* = \arg\min_{\pi \in \Pi} L(\pi, \mathcal{D}). \tag{4}$$

A popular choice for L is the Kullback–Leibler divergence [37] between the distribution of observations induced by the learner policy and teacher policy [38]. We should however take into consideration that demonstrations are costly, since they can be time consuming for the human and ideally we would like the system to have a high level of autonomy to maximize efficiency. Therefore, we wish to optimize task success while minimizing the number of demonstrations:

$$\begin{aligned} &\max_{\mathcal{D}} \mathbb{E}_{\boldsymbol{s}_0 \sim p(\boldsymbol{s}_0)}[V_{\hat{\pi}^*}(\boldsymbol{s}_0)] - \lambda |\mathcal{D}| \\ &\text{s.t.} \quad \hat{\pi}^* = \arg\min_{\pi \in \Pi} L(\pi, \mathcal{D}), \end{aligned} \tag{5}$$

where $|\mathcal{D}|$ is the cardinality of $\mathcal{D}$ (the number of demonstrated trajectories) and λ a regularization parameter. In this way, we have arrived at an expression for the optimal set of demonstrations. The maximization of the expectation of the state values in (5) ensures that we maximize task success, while the regularization term penalizes the number of demonstrations in our dataset. We choose to penalize the number of trajectories $\boldsymbol{\tau}$ rather than their length, since a longer demonstration is preferred over multiple short ones, both by the human and considering the potential cost per demonstration for preparing and processing it.

Finding the optimal solution to the optimization problem in (5) is difficult, because the problem does not have an analytical solution or gradient, and it can quickly become intractable, because it may take a large number of evaluations to find a good solution. In the context of the optimization problem in (5), evaluating the performance of a candidate solution (i.e., a dataset of demonstrations) requires collecting human demonstrations and physical experimentation to evaluate the performance, which can be time-consuming and expensive. Then, it is important to design an optimization algorithm that balances exploration and exploitation and is efficient in terms of the number of evaluations required to find a good solution.

Simulation is a cost-effective means of evaluating candidate solutions without the risks and expenses associated with real-world experimentation. Recent advancements in parallelized physics simulation on accelerated platforms have enabled fast simulations [39], [40]. However, the dissimilarities between the simulator and real-world environments can hinder the transferability of policies learned in simulation to real-world settings. As learning methods tend to exploit these differences to maximize simulated rewards, simulators' conventional use results in overestimation of real-world performance. This may lead to safety hazards and unexpected failures.

To address this issue, we propose to swap the real-world and simulator's roles to synthesize policies using human demonstrations and evaluate them using accelerated physics simulation. By doing so, discrepancies between simulation and reality lead to an underestimation of real-world performance. Failures in simulation may be attributed to either a sub-optimal policy or discrepancies. In case of success, the policy was robust enough to achieve the goal despite the discrepancies. The proposed solution does not rely on gradient information and instead uses a simulation based search strategy to find a good solution for the problem in (5). The system comprises two phases: pre-training and lifelong learning. During the pre-training phase, real-world demonstrations are continuously provided by the human operator, until a certain success rate is achieved in simulation or a maximum number of iterations is reached (see Algorithm 1). After the pre-training phase, the robot enters the lifelong learning phase, where it executes the policy independently. However, it will halt and request a human-demonstration if it encounters a state that resulted in a low success rate in simulation (see Algorithm 2).

Algorithms 1 and 2 provide the workflow of these two phases, while the involved functions are detailed below.

`evaluate`: this function estimates the policy's performance by computing the value function V_π, as defined in (1), which reflects the policy's performance over the induced state distribution, given a reward function R and the initial state distribution $p(\boldsymbol{s}_0)$. Computing V_π may be impractical due to the large number of real-world evaluations required and limited access to the full state $\boldsymbol{s}$. Monte Carlo sampling and an accelerated physics simulator can overcome such challenges (e.g., [39], [40]), as both are suitable for parallel implementation, to estimate an approximate value function $\hat{V}_\pi(\boldsymbol{o})$ that is a function of the observations $\boldsymbol{o}$, serving as a proxy for the real value function. If there are significant differences between the real-world and simulator, a policy trained with real-world demonstrations may, in some cases, always fail to solve the task in simulation. In this case, we propose to use on-policy RL algorithms such as [41].

Algorithm 1: Pre-training phase.

Input: Reward function: $R(\boldsymbol{s})$ // See Eq. (3)
LfD method: $L(\pi, \mathcal{D})$ // See Eq. (4)
Initial states: $\boldsymbol{s}_0 \sim p(\boldsymbol{s}_0)$
Initial policy: π_0
Success threshold: α
Output: Pre-trained policy: π
Dataset: $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$ // Initialize empty dataset
$\pi \leftarrow \pi_0$ // Initialize policy
$\hat{V}_\pi \leftarrow$ `evaluate` $(R, p(\boldsymbol{s}_0), \pi)$ // In simulation
do
  $\boldsymbol{s}_{\text{start}} \leftarrow$ `suggest` $(\hat{V}_\pi)$ // Start state of demo
  $\boldsymbol{\tau} \leftarrow$ `demonstrate` $(\boldsymbol{s}_{start})$ // In real-world
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{\tau}\}$ // Aggregate data
  $\pi \leftarrow$ `optimize` $(L, \mathcal{D})$ // Update policy
  $\hat{V}_\pi \leftarrow$ `evaluate` $(R, p(\boldsymbol{s}_0), \pi)$ // In simulation
until $\mathbb{E}_{\boldsymbol{s}_0 \sim p(\boldsymbol{s}_0)}[\hat{V}_\pi(O(\boldsymbol{s}_0))] > \alpha$ // Success rate

Algorithm 2: Lifelong learning phase.

Input: Reward function: $R(\boldsymbol{s})$ // See Eq. (3)
LfD method: $L(\pi, \mathcal{D})$ // See Eq. (4)
Initial states: $\boldsymbol{s}_0 \sim p(\boldsymbol{s}_0)$
Pre-trained policy: π // See Alg. 1
Dataset: $\mathcal{D}$ // See Alg. 1

$\hat{V}_\pi \leftarrow$ `evaluate` $(R, \mathcal{S}_{start}, \pi)$ // In simulation
$\boldsymbol{s} \leftarrow$ `reset` $(p(\boldsymbol{s}_0))$ // In Real-world
while *running* **do**
  $\boldsymbol{o} \leftarrow O(\boldsymbol{s})$ // Read sensor observations
  if $\hat{V}_\pi(\boldsymbol{o}) > \alpha$ **then** run
    $\boldsymbol{a} \leftarrow \pi(\boldsymbol{o})$ // Get action
    $\boldsymbol{s} \leftarrow$ `act` $(\boldsymbol{a})$ // In real-world
  else request demonstration
    $\boldsymbol{s}_{start} \leftarrow \boldsymbol{s}$ // Demo from current state
    $\boldsymbol{\tau} \leftarrow$ `demonstrate` $(\boldsymbol{s}_{start})$ // In real-world
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{\boldsymbol{\tau}\}$ // Aggregate data
    $\pi \leftarrow$ `optimize` $(L, \mathcal{D})$ // Update policy
    $\hat{V}_\pi \leftarrow$ `evaluate` $(R, p(\boldsymbol{s}_0), \pi)$ // In simulation
  if *done* **then**
    $\boldsymbol{s} \leftarrow$ `reset` $(p(\boldsymbol{s}_0))$ // In real-world

`suggest`: This function proposes a new starting state $\boldsymbol{s}_{\text{start}}$ for the next demonstration based on the policy's simulated performance, which is reflected by the estimated value function $\hat{V}_\pi(\boldsymbol{o})$. Note that the starting state does not necessarily have to lie in the set of initial states $\mathcal{S}_0$. Proper selection of starting states can significantly reduce the number of demonstrations needed to achieve adequate performance, a task that is typically performed by an expert user with knowledge of the chosen LfD method. Alternatively, meta learning, a technique for learning to learn and adapt to new tasks, can be employed [42]. In this case, a meta learning model would be trained to suggest the next starting state for a demonstration expected to improve the current value function $\hat{V}_\pi(\boldsymbol{o})$ the most.

`demonstrate`: This function requests the human to provide a human-demonstration from a given starting state $\boldsymbol{s}_{\text{start}}$. Depending on the task, this could occur via, for example, kinesthetic teaching or teleoperation [22]. Then, `optimize` estimates a policy $\hat{\pi}^*$, as defined in (4), by minimizing the loss function determined by the selected method. Finally, `reset` resets the real-world system to an initial state , and `act` applies the action proposed by the robot's policy π to the system.

IV. Conclusions

This paper conceptually introduced EVA, a system conceived to make the IIL demonstrations process less costly and more safe for humans, while optimizing the exploration and exploitation balance for good generalization.

Motivations for the potential relevance of EVA as a support to IIL applications have been laid down, by analysing state-of-the-art works in the field. The provided core algorithms bring forward the system working flow. Future works shall fully implement EVA, and investigate its actual functionality by assessing its performance with respect to other IIL methods.

References

[1] A. Bonci, P. D. Cen Cheng, M. Indri, G. Nabissi, and F. Sibona, "Human-robot perception in industrial environments: A survey," *Sensors*, vol. 21, no. 5, p. 1571, 2021.

[2] G. Castañé, A. Dolgui, N. Kousi, B. Meyers, S. Thevenin, E. Vyhmeister, and P.-O. Östberg, "The ASSISTANT project: AI for high level decisions in manufacturing," *International Journal of Production Research*, pp. 1–19, 2022.

[3] "NVIDIA Omniverse," Available online: https://developer.nvidia.com/nvidia-omniverse.

[4] P. K. R. Maddikunta, Q.-V. Pham, B. Prabadevi, N. Deepa, K. Dev, T. R. Gadekallu, R. Ruby, and M. Liyanage, "Industry 5.0: A survey on enabling technologies and potential applications," *Journal of Industrial Information Integration*, vol. 26, p. 100257, 2022.

[5] D. Ramesh Kumar, S. Devadasan, and D. Elangovan, "Mapping of non-value adding activities occurring in classical manufacturing companies with lean strategies," *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, vol. 236, no. 3, pp. 894–906, 2022.

[6] E. Johns, "Back to reality for imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 1764–1768.

[7] M. Arduengo, A. Colomé, J. Borràs, L. Sentis, and C. Torras, "Task-adaptive robot learning from demonstration with gaussian process models under replication," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 966–973, 2021.

[8] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[9] Y. Wang, Y. Hu, S. El Zaatari, W. Li, and Y. Zhou, "Optimised learning from demonstrations for collaborative robots," *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102169, 2021.

[10] J. Zhu, M. Gienger, and J. Kober, "Learning task-parameterized skills from few demonstrations," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4063–4070, 2022.

[11] M. Akbulut, E. Oztop, M. Y. Seker, X. Hh, A. Tekden, and E. Ugur, "Acnmp: Skill transfer and task extrapolation through learning from demonstration and reinforcement learning via representation sharing," in *Conference on Robot Learning*. PMLR, 2021, pp. 1896–1907.

[12] Y. Wang, C. C. Beltran-Hernandez, W. Wan, and K. Harada, "An adaptive imitation learning framework for robotic complex contact-rich insertion tasks," *Frontiers in Robotics and AI*, p. 414, 2022.

[13] Y. Lee, A. Szot, S.-H. Sun, and J. J. Lim, "Generalizable imitation learning from observation via inferring goal proximity," *Advances in neural information processing systems*, vol. 34, pp. 16 118–16 130, 2021.

[14] T. Xue, H. Girgin, T. S. Lembono, and S. Calinon, "Guided optimal control for long-term non-prehensile planar manipulation," *arXiv preprint arXiv:2212.12814*, 2022.

[15] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[16] M. Sakr, Z. J. Li, H. M. Van der Loos, D. Kulić, and E. A. Croft, "Quantifying demonstration quality for robot learning and generalization," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9659–9666, 2022.

[17] S. Zhang, Z. Cao, D. Sadigh, and Y. Sui, "Confidence-aware imitation learning from demonstrations with varying optimality," *Advances in Neural Information Processing Systems*, vol. 34, pp. 12 340–12 350, 2021.

[18] L. Panchetti, J. Zheng, M. Bouri, and M. Mielle, "Team: a parameter-free algorithm to teach collaborative robots motions from user demonstrations," *arXiv preprint arXiv:2209.06940*, 2022.

[19] X. Bian, O. Mendez, and S. Hadfield, "Generalizing to new tasks via one-shot compositional subgoals," *arXiv preprint arXiv:2205.07716*, 2022.

[20] B. Hertel and S. R. Ahmadzadeh, "Similarity-aware skill reproduction based on multi-representational learning from demonstration," in *2021 20th International Conference on Advanced Robotics (ICAR)*. IEEE, 2021, pp. 652–657.

[21] R. Burlizzi, M. Vochten, J. De Schutter, and E. Aertbeliën, "Extending extrapolation capabilities of probabilistic motion models learned from human demonstrations using shape-preserving virtual demonstrations," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 772–10 779.

[22] C. Celemin, R. Pérez-Dattari, E. Chisari, G. Franzese, L. de Souza Rosa, R. Prakash, Z. Ajanović, M. Ferraz, A. Valada, J. Kober *et al.*, "Interactive imitation learning in robotics: A survey," *Foundations and Trends® in Robotics*, vol. 10, no. 1-2, pp. 1–197, 2022.

[23] C. Celemin and J. Kober, "Knowledge-and ambiguity-aware robot learning from corrective and evaluative feedback," *Neural Computing and Applications*, pp. 1–19, 2023.

[24] J. Luijkx, Z. Ajanovic, L. Ferranti, and J. Kober, "Partnr: Pick and place ambiguity resolving by trustworthy interactive learning," *arXiv preprint arXiv:2211.08304*, 2022.

[25] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[26] J. Zhang and K. Cho, "Query-efficient imitation learning for end-to-end autonomous driving," *arXiv preprint arXiv:1605.06450*, 2016.

[27] R. Hoque, A. Balakrishna, C. Putterman, M. Luo, D. S. Brown, D. Seita, B. Thananjeyan, E. Novoseller, and K. Goldberg, "Lazydagger: Reducing context switching in interactive imitation learning," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 502–509.

[28] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg, "Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning," *arXiv preprint arXiv:2109.08273*, 2021.

[29] D. K. Jha, S. Jain, D. Romeres, W. Yerazunis, and D. Nikovski, "Generalizable human-robot collaborative assembly using imitation learning and force control," *arXiv preprint arXiv:2212.01434*, 2022.

[30] H. Hu, X. Yang, and Y. Lou, "A robot learning from demonstration framework for skillful small parts assembly," *The International Journal of Advanced Manufacturing Technology*, vol. 119, no. 9-10, pp. 6775–6787, 2022.

[31] G. Franzese, A. Mészáros, L. Peternel, and J. Kober, "Ilosa: Interactive learning of stiffness and attractors," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7778–7785.

[32] A. Mészáros, G. Franzese, and J. Kober, "Learning to pick at non-zero-velocity from interactive demonstrations," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6052–6059, 2022.

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

[34] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[35] A. D. Laud, *Theory and application of reward shaping in reinforcement learning*. University of Illinois at Urbana-Champaign, 2004.

[36] J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine, "How to train your robot with deep reinforcement learning: lessons we have learned," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 698–721, 2021.

[37] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[38] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters *et al.*, "An algorithmic perspective on imitation learning," *Foundations and Trends® in Robotics*, vol. 7, no. 1-2, pp. 1–179, 2018.

[39] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[40] C. D. Freeman, E. Frey, A. Raichuk, S. Girgin, I. Mordatch, and O. Bachem, "Brax–a differentiable physics engine for large scale rigid body simulation," *arXiv preprint arXiv:2106.13281*, 2021.

[41] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[42] F. Alet, T. Lozano-Pérez, and L. P. Kaelbling, "Modular meta-learning," in *Conference on robot learning*, 2018, pp. 856–868.