HiER: Highlight Experience Replay for Boosting Off-Policy Reinforcement Learning Agents

Dániel Horváth, Jesús Bujalance Martín, Ferenc Gábor Erdos, Zoltán Istenes, Fabien Moutarde

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

HiER: Highlight Experience Replay for Boosting Off-Policy Reinforcement Learning Agents

DÁNIEL HORVÁTH[1,2,3], (Member, IEEE), JESÚS BUJALANCE MARTÍN[1], FERENC GÁBOR ERDOS[2], ZOLTÁN ISTENES[3], and FABIEN MOUTARDE[1].

[1]Center for Robotics, MINES Paris, PSL University, Paris, France
[2]Centre of Excellence in Production Informatics and Control, Institute for Computer Science and Control, Hungarian Research Network, Budapest, Hungary
[3]CoLocation Center for Academic and Industrial Cooperation, Eötvös Loránd University, Budapest, Hungary

Corresponding author: Dániel Horváth (e-mail: daniel.horvath@sztaki.hu).

This work was supported in part by the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory and in part by the European Commission through the H2020 project EPIC (https://www.centre-epic.eu/) under grant No. 739592. The work of Dániel Horváth was supported by the Government of France and the Government of Hungary in the framework of "Campus France Bourse du gouvernement français - Bourse Excellence Hongrie".

ABSTRACT Even though reinforcement-learning-based algorithms achieved superhuman performance in many domains, the field of robotics poses significant challenges as the state and action spaces are continuous, and the reward function is predominantly sparse. Furthermore, on many occasions, the agent is devoid of access to any form of demonstration. Inspired by human learning, in this work, we propose a method named highlight experience replay (HiER) that creates a secondary highlight replay buffer for the most relevant experiences. For the weights update, the transitions are sampled from both the standard and the highlight experience replay buffer. It can be applied with or without the techniques of hindsight experience replay (HER) and prioritized experience replay (PER). Our method significantly improves the performance of the state-of-the-art, validated on 8 tasks of three robotic benchmarks. Furthermore, to exploit the full potential of HiER, we propose HiER+ in which HiER is enhanced with an arbitrary data collection curriculum learning method. Our implementation, the qualitative results, and a video presentation are available on the project site: http://www.danielhorvath.eu/hier/

INDEX TERMS Curriculum learning, experience replay, reinforcement learning, and robotics.

I. INTRODUCTION

A high degree of transferability is essential to create universal robotic solutions. While transferring knowledge [1], [2] between domains [3]–[6], robotic systems [7], or tasks [8] is fundamental, it is essential to create and apply universal methods such as reinforcement-learning-based algorithms (RL) [9]–[11] which are inspired by the profoundly universal trial-and-error-based human/animal learning.

RL methods, especially combined with neural networks (deep reinforcement learning), were proven to be superior in many fields such as achieving superhuman performance in chess [12], Go [13], or Atari games [14]. Nevertheless, in the field of robotics, there are significant challenges yet to overcome. Most importantly, the state and action spaces are continuous which intensifies the challenge of exploration. Oftentimes, discretization is not feasible due to loss of information or accuracy, preventing the application of tabular RL methods with high stability. Furthermore, the reward functions of robotic tasks are predominantly sparse which escalates the difficulty of exploration.

Introducing prior knowledge in the form of reward shaping could facilitate the exploration by guiding the agent toward the desired solution. However, 1) constructing a sophisticated reward function requires expert knowledge, 2) the reward function is task-specific, and 3) the agent might learn undesired behaviors. Another source of prior knowledge could be in the form of expert demonstrations. However, collecting demonstrations is oftentimes expensive (time and resources) or even not feasible. Furthermore, it constrains transferability as demonstrations are task-specific.

In parallel to constructing more efficient RL algorithms such as state-of-the-art actor-critic models (DDPG [15],

[16], TD3 [17], and SAC [18]), another line of research focuses on improving existing RL algorithms by controlling the data collection [19]–[27] or the data exploitation [28]–[33] process. Following [34], in this work, we consider both the data collection and the data exploitation methods as curriculum learning (CL) methods [34]–[36]. The former is oftentimes referred to as 'traditional' and the latter as 'implicit' CL.

Our aim is to improve the training of off-policy reinforcement learning agents, particularly in scenarios with continuous state and action spaces, sparse rewards, and the absence of demonstrations. These conditions pose significant challenges for state-of-the-art RL algorithms, due to the challenging problem of exploration.

1) **HiER**: The highlight experience replay creates a secondary experience replay buffer to store the most relevant transitions. At training, the transitions are sampled from both the standard experience replay buffer and the highlight experience replay buffer. It can be added to any off-policy RL agent and applied with or without the techniques of hindsight experience replay (HER) [32] and prioritized experience replay (PER) [28]. If only positive experiences are stored in its buffer, HiER can be viewed as a special, automatic demonstration generator as well. Following [34], HiER is classified as a data exploitation or implicit curriculum learning method.
2) **HiER+**: The enhancement of HiER with an arbitrary data collection (traditional) curriculum learning method. The overview of HiER+ is depicted in Fig.1. Furthermore, as an example of the data collection CL method, we propose E2H-ISE, a universal, easy-to-implement *easy2hard* data collection CL method that requires minimal prior knowledge and controls the entropy of the initial state-goal distribution $\mathcal{H}(\mu_0)$ which indirectly controls the task difficulty[1]

To demonstrate the universality of our methods, HiER is validated on 8 tasks of three different robotic benchmarks [37]–[39] based on two different simulators [40], [41], while HiER+ is evaluated on the push, slide, and pick-and-place tasks of the Panda-Gym [37] robotic benchmark. Our methods significantly improve the performance of the state-of-the-art algorithms for each task.

The paper is structured as follows: in Section II and III, the essentials of RL and CL are described followed by a literature review. In Section IV and V, HiER, E2H-ISE, and HiER+ are presented with the experimental results. Finally, the summary of our findings is provided in Section VI.

II. BACKGROUND

A. REINFORCEMENT LEARNING

In reinforcement learning, an agent attempts to learn the optimal policy for a task through interactions with an environment. It can be formalized with a Markov decision process represented by the state space $\mathcal{S}$, the action space $\mathcal{A}$, the transition probability $p(s_{t+1}|s_t, a_t)$, where $s \in \mathcal{S}$ and $a \in \mathcal{A}$, the reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, the discount factor $\gamma \in [0, 1]$, and the initial state distribution μ_0 [9].

Every episode starts by sampling from the initial state distribution μ_0. In every timestep $t \in \mathbb{N}$, the agent performs an action according to its policy $\pi(a|s)$ and receives a reward, a new state[2], and a done flag[3] $d \in \{0, 1\}$ from the environment. In the case of off-policy algorithms, the $(s_t, a_t, s_{t+1}, r_t, d_t)$ tuples called transitions are stored in the so-called experience replay buffer $\mathcal{B}_{er}$ which is a circular buffer and the batches for the weight updates are sampled from it.

Learning the optimal policy is formulated as maximizing the expected discounted sum of future rewards or expected return $\mathbb{E}_{s_0}[R_0^{disc}|s_0]$ and $R_t^{disc} = \sum_{i=t}^{T} \gamma^{i-t} r_i$, where $T \in \mathbb{N}$ is the time horizon. Value-based off-policy algorithms learn the optimal policy by learning the optimal Q (action-value) function: $Q^{\pi}(s_t, a_t) = \mathbb{E}[R_t^{disc}|s_t, a_t]$.

In multi-goal tasks, there are multiple reward functions r^g parametrized by the goal $g \in \mathcal{G}$. A goal is described with a set of states $\mathcal{S}^g \subset \mathcal{S}$, and it is achieved when the agent is in one of its goal states $s_t \in \mathcal{S}^g$ [25]. Thus, according to [42] and [25], the policy is conditioned also on the goal $\pi(a|s, g)$. In our implementation, we simply insert goal g into state s and consequently, when the initial state is sampled from μ_0, the goal is sampled as well. Henceforth, we refer to μ_0 as the initial state-goal distribution.

In robotics, sparse reward function is often formulated as:

$$r(s, \cdot) = \begin{cases} 0, & \text{if } s \in \mathcal{S}^g \\ -1, & \text{otherwise} \end{cases} \tag{1}$$

Another important aspect of an RL task is whether the agent has access to any form of demonstration. A demonstration is an example of the desired (optimal or suboptimal) behavior provided by an external source which can significantly facilitate the exploration [43]. Oftentimes, an expert human provides these examples in which case it can be referred to as human demonstrations. Nevertheless, collecting expert demonstrations is expensive and time-consuming, or even not feasible. On the other hand, automatically generating demonstrations presumes that the task can be solved already, which raises the question of why RL training is needed in the first place. For the aforementioned reasons, in this paper, we assume that the agent is devoid of access to any form of demonstration.

B. CURRICULUM LEARNING

In this section, the field of CL is briefly presented. For a thorough overview, we refer the reader to [34], [36].

CL, introduced by Bengio et al. [35], attempts to facilitate the machine-learning training process. Similar to how

[1]ISE stands for *initial state entropy*.

[2]For simplicity, the environment is considered to be fully observable.

[3]Indicating the end of the episode.

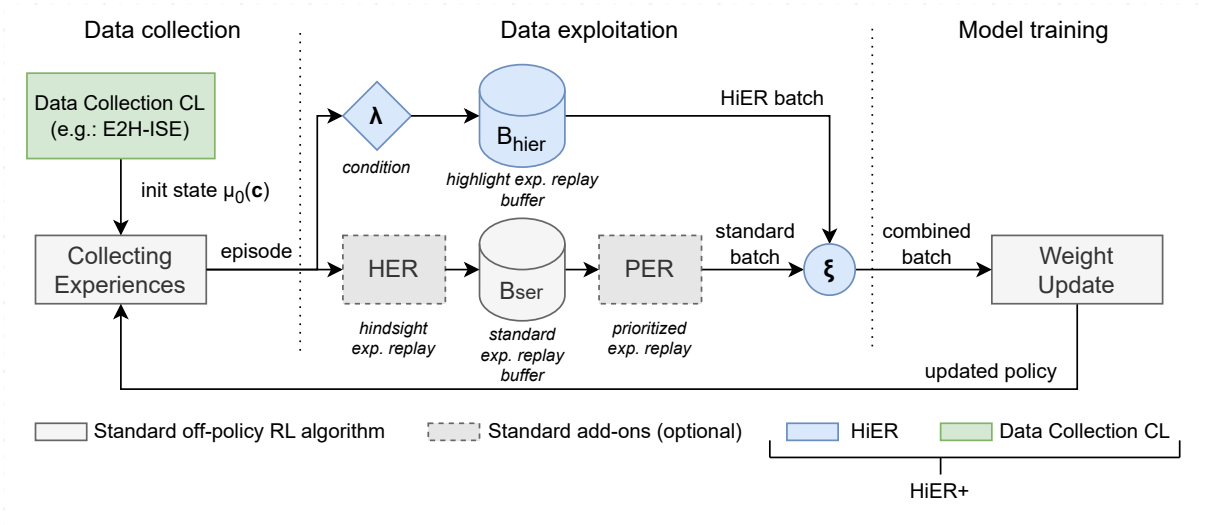


FIGURE 1: The overview of HiER and HiER+. For every episode, the initial state is sampled from μ_0. After every episode, the transitions are stored in $\mathcal{B}_{ser}$, and in case the λ condition is fulfilled then in $\mathcal{B}_{hier}$ as well. For training, the transitions are sampled from both $\mathcal{B}_{ser}$ and $\mathcal{B}_{hier}$ according to the ratio ξ. For a detailed description, see Alg. 1.

humans require a highly-organized training process (introducing different concepts at different times) to become fully-functional adults, machine-learning-based models might as well benefit from a similar type of curriculum.

Originally, the curriculum followed an *easy2hard* or *starting small* structure [35], however, conflicting results with hard example mining [44] led to a more general definition of CL which did not include the *easy2hard* constraint.

In supervised learning, a CL framework typically consists of two main components: the difficulty measurer and the training scheduler. The former assigns a difficulty score to the samples, while the latter arranges which samples can be used and when for the weight updates.

According to [34], in reinforcement learning, CL can typically control either the data collection or the data exploitation process. The data collection process can be controlled by changing the initial state distribution, the reward function, the goals, the environment, or the opponent. The data exploitation process can be controlled by transition selection or transition modification. HiER belongs to the data exploitation branch of CL while E2H-ISE is classified as a data collection CL method.

C. EVALUATION METHODS

State-of-the-art deep reinforcement learning models are compared based on just a few experiments, primarily due to constraints on training time. Therefore, simple point estimates of aggregate performance such as mean and median scores across tasks are insufficient as they do not capture the statistical uncertainty implied by the finite number of training runs. In this section, we present the most relevant statistical evaluation methods utilized in RL.

In general, confidence intervals (CIs) are beneficial to measure uncertainty. The bootstrap CI method creates multiple datasets by resampling with replacement from a set of data points (results of independent training runs). As the distribution of the means of the resampled datasets approaches a normal distribution [4], the CI can be calculated. Traditionally, bootstrap CI is performed on a single task [45]–[47]. Agarwal et al. [48] proposes the method of stratified bootstrap CI which performs a bootstrap CI across multiple tasks using stratified sampling.

Another useful evaluation method is presenting the performance profiles. A tail distribution function is defined as $F(\tau) = P(X > \tau)$, where $\tau \in \mathbb{R}$, and X is a real-valued random variable[5]. The performance profiles are beneficial for comparing different algorithms at a glance. In mathematical terms, X has stochastic dominance over Y if $P(X > \tau) \geq P(Y > \tau)$, for all τ, and for some τ $P(X > \tau) > P(Y > \tau)$, where X and Y are random variables. Two main versions are the run-score distribution. [48] and the average-score distributions [49]. Examples of performance profiles are presented in Fig. 4 and the left side of Fig. 8.

Displaying the probability of improvement is another beneficial evaluation method. It shows the probability of Algorithm X exceeding Algorithm Y in a set of tasks. Important to note that it only indicates the probability of improvement and not the magnitude of the improvement.

Finally, standard aggregate performance metrics have shortcomings. The median has high variability and it is unchanged even when half of the results are zero, while the mean can be significantly influenced by some outliers. Thus, [48] proposes the interquartile mean (IQM) and the optimality gap (OG) as alternatives to the median and the

[4]Central limit theorem.

[5]Performance estimates are random variables, based on a finite number of runs.

mean. IQM removes the bottom and top 25% of the runs and calculates the mean of the remaining 50% of the runs. The OG represents the shortfall of the algorithm in achieving a desirable target. It is important to note that the extent to which an algorithm surpasses the desired target does not affect its OG score.

III. RELATED WORKS

A. DATA EXPLOITATION

Schaul et al. [28] proposed the technique of prioritized experience replay (PER) which controls the transition selection by assigning priority (importance) scores to the samples of the replay buffer based on their last TD error [50] and thus, instead of uniformly, they are sampled according to their priority. Additionally, as high-priority samples would bias the training, importance sampling is applied.

As a form of prioritization, Oh et al. [29] introduced self-imitation learning (SIL) for on-policy RL. The priority is computed based on the discounted cumulative rewards. Furthermore, the technique of clipped advantage is utilized to incentivize positive experiences. By modifying the Bellmann optimality operator, Ferret et al. [30] introduced self-imitation advantage learning which is a generalized version of SIL for off-policy RL.

Wang et al. [31] presented the method of emphasizing recent experience which is a transition selection technique for off-policy RL agents. It prioritizes recent data without forgetting the past while ensuring that updates of new data are not overwritten by updates of old data.

Andrychowicz et al. [32] introduced the technique of hindsight experience replay (HER) which performs transition modification to augment the replay buffer by adding virtual episodes. After collecting an episode and adding it to the replay buffer, HER creates virtual episodes by changing the (desired) goal to the achieved goal at the end state (or to another state depending on the strategy) and relabeling the transitions before adding them to the replay buffer.

Bujalance and Moutarde [33] propose reward relabeling to guide exploration in sparse-reward robotic environments by giving bonus rewards for the last L transitions of the episodes.

B. DATA COLLECTION

Florensa et al. [21] presented the reverse curriculum generation method to facilitate exploration for model-free RL algorithms in sparse-reward robotic scenarios. At first, the environment is initialized close to the goal state. For new episodes, the distance between the initial state and the goal state is gradually increased. As prior knowledge, at least one goal state is required. To sample 'nearby' feasible states, the environment is initialized in a certain seed state (in the beginning at a goal state), and then, for a specific time, random Brownian motion is executed.

Ivanovic et al. [22] proposed the backward reachability curriculum method which is a generalization of [21] utilizing prior knowledge of the simplified, approximate dynamics of the system. They compute the approximate backward reaching sets using the Hamilton-Jacobi reachability formulation and sample from them using rejection sampling.

Salimans and Chen [23] facilitate exploration by utilizing one human demonstration. In their method, the initial states come from the demonstration. More precisely, until a timestep $t_D \in \mathbb{N}$, the agent copies the actions of the demonstration, and after t_D, it takes actions according to its policy. During the training, t_D is moved from the end of the demonstration to the beginning of the demonstration. Their method outperformed state-of-the-art methods in the Atari game Montezuma's Revenge. Nevertheless, arriving at the same state after a specific sequence of actions (as in the demonstration) is rather unlikely, especially when the transition function is profoundly stochastic, such as in robotics.

Sukhbaatar et al. [24] present automatic curriculum generation with asymmetric self-play of two versions of the same agent. One proposes tasks for the other to complete. With an appropriate reward structure, they automatically create a curriculum for exploration.

Florensa et al. [25] create a curriculum for multi-goal tasks by sampling goals of intermediate difficulty. First, the goals are labeled based on their difficulty, and then a generator is trained to output new goals with appropriate difficulty to efficiently train the agent.

Pong et al. [26] proposed Skew-Fit, an automatic curriculum that attempts to create a better coverage of the state space by maximizing the entropy of the goal-conditioned visited states $\mathcal{H}(\mathcal{S}|\mathcal{G})$ by giving higher weights to rare samples. Skew-Fit converges to uniform distribution under specific conditions.

Racanière et al. [27] proposed an automatic curriculum generation method for goal-oriented RL agents by training a setter agent to generate goals for the solver agent considering goal validity, goal feasibility, and goal coverage.

IV. METHOD

In this Section, our contributions are presented. First, HiER in Section IV-A, and then E2H-ISE and HiER+ in Section IV-B and IV-C. Our implementation is available at our git repository[6].

A. HIER

Humans remember certain events stronger than others and tend to replay them more frequently than regular experiences thus learning better from them [51]. As an example, an encounter with a lion or scoring a goal at the last minute will be engraved in our memory. Inspired by this phenomenon, HiER attempts to find these events and manage them differently than regular experiences. In this paper, only positive experiences are considered with HiER, thus it can be viewed as a special, automatic demonstration generator as well.

[6]https://github.com/sztaki-hu/hier

PER and HER control what transitions to store in the experience replay buffer and how to sample from them. Contrary to them, HiER creates a secondary experience replay buffer. Henceforth, the former buffer is called standard experience replay buffer $\mathcal{B}_{ser}$, and the latter is referred to as highlight experience replay buffer $\mathcal{B}_{hier}$. At the end of every episode, HiER stores the transitions in $\mathcal{B}_{hier}$ if certain criteria are met. For updates, transitions are sampled both from the $\mathcal{B}_{ser}$ and $\mathcal{B}_{hier}$ based on a given sampling ratio. HiER is depicted in Fig. 1 marked in blue.

The criteria can be based on any type of performance measure, in our case, the undiscounted sum of rewards $R = \sum_{i=0}^{T} r_i$ was chosen. The reward function r is formulated as in Eq. (1). Although more complex criteria are possible, in this work, we consider only one performance measure and one criterion: if R is greater than a threshold $\lambda \in \mathbb{R}$ then all the transitions of that episode are stored in $\mathcal{B}_{hier}$ and $\mathcal{B}_{ser}$, otherwise only in $\mathcal{B}_{ser}$. Nevertheless, λ can change in time, thus we define a λ_j for every j where $j \in \mathbb{N}$ is the index of the episode. In this work, the following λ modes were considered:

- `fix`: $\lambda_j = Z_\lambda$ for every j where $Z_\lambda \in \mathbb{R}$ is a constant.[7]
- `predefined`: λ is updated according to a predefined profile. Profiles could be arbitrary, such as linear, square-root, or quadratic. In this work only the linear profile with saturation was considered:

$$\lambda_j = \min\left(1,\ \frac{t}{T_{total} \cdot z_{sat}}\right) \tag{2}$$

  where $t \in \mathbb{N}$ and $T_{total} \in \mathbb{N}$ are the actual, and the total timesteps of the training and $z_{sat} \in [0,1]$ is a scaler, indicating the start of the saturation.[8]
- `ama` (adaptive moving average): λ is updated according to:

$$\lambda_j = \begin{cases} \min\left(\lambda_{max},\ M + \frac{1}{w}\sum_{i=1}^{w} R_{j-i}\right), & \text{if } j > w \\ \lambda_0, & \text{otherwise} \end{cases} \tag{3}$$

  where $\lambda_0 \in \mathbb{R}$ is the initial value of λ, while $\lambda_{max} \in \mathbb{R}$ is the maximum value allowed for λ. Furthermore, $w \in \mathbb{Z}^+$ is the window size and $M \in \mathbb{R}$ is a constant shift.[9]

Another relevant aspect of HiER is the sampling ratio between $\mathcal{B}_{ser}$ and $\mathcal{B}_{hier}$ for weight update, defined by $\xi \in [0,1]$. It can change in time, updated after every weight update, thus we define a ξ_k for every k where $k \in \mathbb{N}$ is the index of the weight update. The following versions were considered:

- `fix`: $\xi_k = Z_\xi$ for every k where $Z_\xi \in \mathbb{R}$ is a constant.
- `prioritized`: ξ is updated according to:[10]

$$\xi_k = \frac{L_{hier,k}^{\alpha_p}}{L_{hier,k}^{\alpha_p} + L_{ser,k}^{\alpha_p}} \tag{4}$$

  where $L_{hier,k} \in \mathbb{R}$ and $L_{ser,k} \in \mathbb{R}$ are the TD errors of the training batches sampled from $\mathcal{B}_{hier}$ and $\mathcal{B}_{ser}$ at k. The parameter $\alpha_p \in [0,1]$ determines how much prioritization is used.[11]

Sampling from $\mathcal{B}_{hier}$ and not only from $\mathcal{B}_{ser}$ introduce a bias towards the experiments collected in $\mathcal{B}_{hier}$. This bias is similar in nature to the case when demonstrations are utilized. In that scenario, the expert demonstrations are sampled and combined with online experience, biasing the exploration towards the desired behavior. In our case, as the agent is devoid of any form of demonstration, $\mathcal{B}_{hier}$ serves similarly as a demonstration buffer. This bias is essential for achieving enhanced performance (presented in Section V). However, some characteristics of the proposed methods mitigate the sampling bias. The `predefined` and the `ama` λ methods alleviate the bias by setting the entry of $\mathcal{B}_{hier}$ lower at the beginning and gradually increasing it resulting in a higher cardinality for $\mathcal{B}_{hier}$ and higher similarity between $\mathcal{B}_{hier}$ and $\mathcal{B}_{ser}$. Furthermore, the presented `prioritized` ξ method prevents overfitting on the data of $\mathcal{B}_{hier}$ as low L_{hier} loss reduces ξ (see Equation 4). On the other hand, the bias could be further reduced by gradually decreasing ξ over time, or the gradient of the data from $\mathcal{B}_{hier}$ could be scaled, similarly to importance sampling in the case of PER [28].

Another relevant aspect worth detailing is the difference between the `prioritized` ξ method and PER. While PER changes the probability distribution of selecting specific transitions from $\mathcal{B}_{ser}$ based on their individual TD error, the `prioritized` ξ method controls sampling between $\mathcal{B}_{ser}$ and $\mathcal{B}_{hier}$ based on the mean TD error of the data selected from $\mathcal{B}_{ser}$ and $\mathcal{B}_{hier}$. Thus, the sampling distribution of PER has $|\mathcal{B}_{ser}|$ outputs while the sampling distribution of the `prioritized` ξ method has two outputs, one for $\mathcal{B}_{ser}$ and one for $\mathcal{B}_{hier}$. Another relevant difference is that in the `prioritized` ξ method, contrary to PER, the gradients are not scaled, similar to a standard demonstration buffer.

Important to note that the formulation of HiER is fundamentally different from [28]–[33], not only but most importantly because of the idea of the secondary experience replay.

B. E2H-ISE

A key attribute of HiER is that it learns from relevant positive experiences, described in Section IV-A. However, if these experiences are scarce in the first place, $\mathcal{B}_{hier}$ would be considerably limited or even empty. Thus, HiER could benefit from an *easy2hard* data collection CL method by having access to more positive experiences.

[7] We also tried a version with n highlight buffers and n thresholds $\lambda_1, \lambda_2, ..., \lambda_n$. An episode is stored in the highlight buffer with the highest λ_i for which $R > \lambda_i$.

[8] In the equation, λ_j does not directly depend on j. However as t increases, so does j and λ_j with it.

[9] In an alternative version M is not a constant but relative to $\frac{1}{w}\sum_{i=1}^{w} R_{j-i}$.

[10] Similarly as in the case of PER.

[11] If $\alpha_p = 0$, then $\xi = 0.5$ regardless $L_{hier,k}$ and $L_{ser,k}$.

E2H-ISE is a data collection CL method based on controlling the entropy of the initial state-goal distribution $\mathcal{H}(\mu_0)$ and with it, indirectly, the task difficulty. In general, μ_0 is constrained to one point (zero entropy) and moved towards the uniform distribution on the possible initial space (max entropy). Even though certain E2H-ISE versions allow decreasing the entropy, in general, they move μ_0 towards max entropy.

To formalize E2H-ISE, the parameter $c \in [0,1]$ is introduced as the scaling factor of the uniform μ_0, assuming that the state space, including the goal space, is continuous and bounded. The visualization of the scaling factor c is depicted in Fig. 2. If $c = 1$ there is no scaling, while $c = 0$ means that μ_0 is deterministic and returns only the center point of the space. To increase or decrease $\mathcal{H}(\mu_0)$, c changes in time, thus we define c_j for every j where $j \in \mathbb{N}$ is the index of the episode. At the start of the training, c is initialized and it is updated at the beginning of every training episode before s_0 is sampled from μ_0.[12] The following versions are proposed for updating c:

- `predefined`: c changes according to a predefined profile similar as in the case of λ `predefined` (see Section IV-A). In this paper, only the linear profile with saturation was considered.[13]
- `self-paced`: c is updated according to:

$$c_j = \begin{cases} \min(1,\ c_{j-1} + \delta), & \text{if } P_{train,w} > \Psi_{high} \\ \max(0,\ c_{j-1} - \delta), & \text{if } P_{train,w} < \Psi_{low} \\ c_{j-1}, & \text{otherwise} \end{cases} \tag{5}$$

  where $P_{train,w} \in \mathbb{R}$ is mean of last $w \in \mathbb{Z}^+$ (window size) training success rate, $\delta \in [0,1]$ is the step size, and $\Psi_{high} \in \mathbb{R}$ and $\Psi_{low} \in \mathbb{R}$ are threshold values.[14] After any update on c, $P_{train,w}$[15] is emptied, and the update on c is restarted after w episodes.
- `control`: c is updated according to:

$$c_j = \begin{cases} \min(1,\ c_{t-j} + \delta), & \text{if } P_{train,w} \geq \Psi \\ \max(0,\ c_{t-j} - \delta), & \text{if } P_{train,w} < \Psi \end{cases} \tag{6}$$

  where $\Psi \in \mathbb{R}$ is the target. The algorithm attempts to move and keep $P_{train,w}$ at Ψ. Updates are executed only if $j > w$.
- `control adaptive`: This method is similar to `control` but the target success rate Ψ is not fixed but computed from the mean evaluation success rate:

$$\Psi_j = \min\left(\Psi_{max},\ \Delta + \frac{1}{w}\sum_{i=1}^{w} R_{j-i}^{eval}\right) \tag{7}$$

  where $\Delta \in [0,1]$ is a constant shift (as we want to target a better success rate than the current) and $\Psi_{max} \in \mathbb{R}$ is the maximum value allowed for Ψ.[16] Updates are executed only if $j > w$.

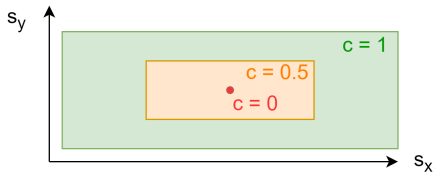


FIGURE 2: Visualization of the effect of parameter c on μ_0 in a 2D case where state $s = [s_x, s_y]$. The initial state $s_0 = [s_{0,x}, s_{0,y}]$ is sampled from the probability distribution $\mu_0(c)$.

Sampling from $\mu_0(c \neq 1)$ introduces bias to the states within the probability distribution of $\mu_0(c)$. This bias is reduced as c increases. Furthermore, as the buffers are circular, once they reach their capacity, the old experiences are replaced with new ones. On the other hand, we conducted experiments on dynamically subtracting the center of $\mu_0(c)$ to counterbalance the sampling bias, e.g.: $\mu_0(c) = \mu_0(c_1) - \mu_0(c_2)$ where $c_1 > c_2$. However, they did not result in any improvement. Our experimental results, presented in Section V, show that accepting the bias and starting with c close to zero is beneficial as HiER+ further improves the performance of HiER.

Important to note, our E2H-ISE formulation is inherently different from [21]–[23] as our solution does not concentrate on goal difficulty but the entropy of μ_0. In our case, the *easy2hard* attribute derives from the magnitude of the entropy and not from the goal difficulty. It is also disparate from [26] as their solution maximizes the entropy of goal-conditioned visited states $\mathcal{H}(\mathcal{S}|\mathcal{G})$ and not $\mathcal{H}(\mu_0)$. Nevertheless, the E2H-ISE method is only an example of data collection CL methods that can be utilized in HiER+. It is proposed in this paper, as it is significantly easier to implement than the presented, more sophisticated, state-of-the-art methods, while it is universal and requires minimal prior knowledge. Thus, the full potential of HiER+ can be presented conveniently with the E2H-ISE method. Comparing different data collection CL methods in HiER+ is out of the scope of this work.

C. HIER+

In this section, HiER+ is presented which is an enhancement of HiER with an arbitrary data collection CL method. Even though in this work, we present HiER+ with E2H-ISE, it is important to note that the fundamental architecture of HiER+ would remain consistent when paired with alternative data collection CL approaches. It can be added to any off-policy RL algorithm with or without HER and

[12]For evaluation, the environment is always initialized according to the unchanged μ_0.

[13]We have experimented with a 2-stage version where μ_0 and μ_G (initial goal distribution) were separated.

[14]If $\Psi_{low} = 0$, then c can only increase.

[15]The circular buffer storing the success rates.

[16]Important to note that contrary to the training, in the evaluation, we sample from the unrestricted μ_0 ($c = 1$), thus the eval success rate represents the real success rate of the agent. Consequently, c can be set to keep the training to a success rate that is just (by Δ) above the eval success rate.

PER, as depicted in Fig. 1 and presented in Algorithm 1. Having initialized the variables and the environment (Lines 1-6), the training loop starts. After collecting an episode, its transitions are stored in $\mathcal{B}_{ser}$, and if HER is active then virtual experiences are added as well (Lines 13-14).[17] Then the λ parameter of HiER is updated and if the given condition is met, the episode is stored in $\mathcal{B}_{hier}$ as well (Lines 15-18). In the next steps, the c parameter of E2H-ISE is updated and the environment is reinitialized (Line 19-21), thus the agent can start collecting the next episode. At a given frequency, the weights of the models are updated (Line 23-31). The batches of $\mathcal{D}_{ser}$ and $\mathcal{D}_{hier}$ are sampled and combined (Lines 24-26). After the weight update (Line 27), if PER is active, the priorities in $\mathcal{B}_{ser}$ are updated (Line 28). Finally, the ξ parameter and with it the batch size of HiER is updated (Lines 29-30).

Algorithm 1 HiER+

1: Initialize $c \leftarrow 0$, λ, ξ, n, θ, ϕ
2: $n_{hier} \leftarrow \xi \cdot n$ ▷ n: batch size
3: Initialize $\mathcal{B}_{ser} \leftarrow \emptyset$
4: Initialize $\mathcal{B}_{hier} \leftarrow \emptyset$
5: Initialize episode buffer $\mathcal{E} \leftarrow \emptyset$
6: $s \leftarrow \mu_0(c_0)$ ▷ Init env
7: **while** Convergence **do**
8: $a \leftarrow \pi_\theta(s)$ ▷ Collecting data
9: $s_2, r, d \leftarrow$ Env.step(a)
10: $\mathcal{E} \leftarrow \mathcal{E} \cup (s, a, s_2, r, d)$
11: $s \leftarrow s_2$
12: **if** Episode ends **then**
13: $\mathcal{B}_{ser} \leftarrow \mathcal{B}_{ser} \cup \mathcal{E}$ ▷ Store transitions of $\mathcal{E}$
14: $\mathcal{B}_{ser} \leftarrow \mathcal{B}_{ser} \cup \mathcal{E}_{virtual}$ ▷ HER (optional)
15: Update λ_j ▷ HiER: Section IV-A
16: **if** $\lambda_j < \sum_{r_i \in \mathcal{E}} r_i$ **then**
17: $\mathcal{B}_{hier} \leftarrow \mathcal{B}_{hier} \cup \mathcal{E}$
18: **end if**
19: Update c_j ▷ E2H-ISE: Section IV-B
20: $\mathcal{E} \leftarrow \emptyset$
21: $s \leftarrow \mu_0(c_j)$
22: **end if**
23: **if** Weight update **then**
24: $\mathcal{D}_{ser} \leftarrow$ select $(n - n_{hier})$ sample from $\mathcal{B}_{ser}$
25: $\mathcal{D}_{hier} \leftarrow$ select n_{hier} sample from $\mathcal{B}_{hier}$
26: $\mathcal{D} \leftarrow \mathcal{D}_{cer} + \mathcal{D}_{hier}$
27: Update weights θ, ϕ based on $\mathcal{D}$
28: Update priorities in $\mathcal{B}_{ser}$ ▷ PER (optional)
29: Update ξ_k ▷ HiER: Section IV-A
30: $n_{hier} \leftarrow \xi_k \cdot n$
31: **end if**
32: **if** Evaluation **then**
33: Evaluate agent with $\mu_0(c = 1)$ ▷ Standard init
34: **end if**
35: **end while**

V. RESULTS

Our contributions were validated on 8 tasks of three robotic benchmarks. The tasks are the push, slide, and pick-and-place tasks of the Panda-Gym [37] and the Gymnasium-Robotics Fetch benchmarks [38], and two mazes, depicted on Fig. 11 of the Gymnasium-Robotics PointMaze environment [39]. The Panda-Gym Environment is based on the PyBullet [40] physics engine while the Gymnasium-Robotics Fetch and PointMaze environments are based on MuJoCo [41].

It is important to note, that for all tasks, the state and action spaces are continuous, and the reward function is sparse without any reward shaping. Furthermore, the agent is devoid of access to any form of demonstration. These constraints, significantly exacerbate the difficulty of exploration.

The naming convention of the algorithms is the following: Algorithm [Components]. The algorithm can be either Baseline, HiER, or HiER+, and the options for the components are HER and PER [18]. Thus, Baseline [HER & PER] means that the base (SAC, TD3, or DDPG) RL algorithm was applied with HER and PER. On the other hand, HiER [HER] means that the base RL algorithm was applied with our HiER method and HER but without PER. HiER+ is HiER with E2H-ISE.

First and foremost, we present our evaluation protocol in Section V-A which is essential for result reproducibility. Then, the aggregate performance (across all tasks) of HiER is shown compared to their corresponding baselines in Section V-B. Subsequently, HiER and HiER+ (with E2H-ISE) are thoroughly evaluated on the push, slide, and pick-and-place tasks of the Panda-Gym robotic benchmark in Section V-C. Furthermore, HiER is evaluated on the push, slide, and pick-and-place tasks and two mazes, depicted on Fig. 11 of the Gymnasium-Robotics Fetch and PointMaze benchmarks in Section V-D and Section V-E. Then, the qualitative results of all tasks are evaluated in Section V-F. Additionally, the comparisons of the different ξ, λ, and c methods are presented in Section V-G1 and V-G2. Finally, our method is validated with DDPG and TD3 in Section V-G3.

For our experiments, the SAC RL algorithm was chosen, except in Section V-G3. The standard hyperparameters are set as default in [52] except the discount factor $\gamma = 0.95$ as in [37], and the SAC entropy maximization term $\alpha = 0.1$. The buffer size of $\mathcal{B}_{hier}$ was set to 10^6.

In all the experiments with the exception of Section V-G1 and V-G2, HiER was applied with the `predefined` λ method and with the `prioritized` version of ξ when PER was active and with the `fix` version with $\xi = 0.5$ otherwise. Furthermore, in HiER+, the E2H-ISE method was employed with the `self-paced` option. The aforementioned settings were selected according to our comparison presented in Section V-G1 and V-G2.

[17] $\mathcal{B}_{ser}$ and $\mathcal{B}_{hier}$ are circular buffers, thus once they are full, the new transitions are replacing the old ones.

[18] With the exception of Fig. 16.

A. EVALUATION PROTOCOL

For results reproducibility, it is important to disclose the evaluation protocol. Each algorithm (configuration) and task pair is trained in 10 independent runs with different random seeds. For every run, at a specified frequency, the evaluation performance of the model is measured, presented at Line 32-34 of Algorithm 1. The two most relevant performance metrics are the evaluation success rate and the evaluation accumulated reward, henceforth success rate and reward. In this paper, the performance is measured 50 times during a single training, and each time, the evaluation score is computed by taking the mean of 100 episodes. At the end of the training, all evaluation data is saved and stored. For the evaluation presented in this paper, in the case of success rates, the best scores of each run were the base datapoints[19], meaning for each algorithm (configuration) and task pair there are 10 datapoints, one for every run. This evaluation protocol follows [14], [53], [54] and the idea is similar to the method of early stopping.

In the following sections, the primary basis of evaluation is the success rate which was chosen for the following reasons:

- Our main objective is to solve the tasks with the highest success rate. As we focus on sparse reward scenarios with Equation 1, the only additional information in the reward score is how fast the agent solved the task which is less relevant in our case.
- The success rate is an already normalized scale between zero and one. Reward scores of Equation 1 with different time horizons are significantly disparate.
- The reward value depends on the reward function itself. The same task can be executed with a different reward function, whose results are not comparable.
- The success rate could be seen as a specific reward function giving zero reward to every non-goal state, and one for every goal state.

Nevertheless, we report our reward scores, for the aggregated results, presented in Tab. 1, and for the results of HiER and HiER+ on the Panda-Gym environment, displayed in Tab. 4. In the cases of reward scores, instead of the best, the last values of each run were utilized. Our aim is to show that our methods outperform the state-of-the-art not only in the chosen evaluation protocol but in other protocols as well.

In general, we present our results with the mean, median, interquartile mean (IQM), and optimality gap (OG) metrics. For the former three, higher values are better, while for OG, the lower score is better. In the case of the success rate, the desired target is 1.0 which is the maximum achievable score[20]. For displaying the amount of uncertainty, in the graphs, 95% confidence intervals (CIs) were applied.

For plotting the figures of aggregated results, the performance profiles, and the probability improvements, the *rliable* [48] library was utilized. Having 10 runs was sufficient, thus we present our results without task bootstrapping (as default in *rliable*).

[19]For calculating the mean, median, IQM, and OG scores.

[20]As the desired target is 1.0 which is in itself the highest possible number, these results are redundant as the mean is also presented. Nevertheless, to facilitate comparison, we preferred to keep them in the graphs.

B. AGGREGATED RESULTS ACROSS ALL TASKS

Prior to showing the experimental results on each of the three robotic benchmarks, this section provides a summary of the aggregated results across all tasks, focusing on HiER and HiER [HER].

Our experimental results are presented in Tab. 1 and Fig. 3. The results indicate that both HiER versions outperform their corresponding baseline, and HiER [HER] yields the best performance in all metrics. In terms of point estimates, while Baseline [HER] yields 0.56 and -43.7 IQM success rate and IQM reward, HiER [HER] achieves 0.83 and -32.48 scores which are increments of 0.27 and 11.22, respectively. Moreover, regarding the uncertainty, both HiER and HiER [HER] are superior to their corresponding baselines as the confidence intervals do not overlap.

Additionally, the performance profile graph, presented in Fig. 4, displays the run-score and the average-score distributions of the aforementioned algorithms. It shows that both HiER and HiER [HER] have stochastic dominance over their baselines.

Finally, Fig. 5 shows that both HiER and HiER [HER] outperform their baselines with 0.85 and 0.88 probability[21]. Additionally, HiER [HER] surpasses HiER with a probability of 0.76.

C. PANDA-GYM

Having presented the aggregated results on HiER, we present our results on the Panda-Gym robotic benchmark with more details and task-specific results. Additionally, we demonstrate how HiER+ with E2H-ISE can further improve the performance of HiER. From the Panda-Gym robotic benchmark, three robotic manipulation tasks were considered:

- `PandaPush-v3`: A block needs to be pushed to a target. Both the block starting position and the target position are within the reach of the robot.
- `PandaSlide-v3`: A puck needs to be slid to a target position outside of the reach of the robot.
- `PandaPickAndPlace-v3`: A block needs to be moved to a target that is oftentimes in the air thus the robot needs to grasp the block.

The starting position of the block (or the puck) and the goal position are sampled from the corresponding distributions. The action space is composed of incremental actions on the tool center point in x, y, and z axes. Furthermore, in the case of the `PandaPickAndPlace-v3` task, the action space expanded with a continuous gripper control action. The reward function is sparse, as described in Eq.(1). The

[21]Important to note, that these probabilities could be significantly higher if the easy tasks were removed.

TABLE 1: HiER compared to the state-of-the-art across all tasks. For the reward, there is no universal desirable target, thus there is no OG value. The column-wise best results are marked in bold. Both HiER version outperform their corresponding baseline. HiER [HER] yields the best performance in all metrics.

	HER	HiER	Success rate				Reward		
			Mean ↑	Median ↑	IQM ↑	OG ↓	Mean ↑	Median ↑	IQM ↑
Baselines	-	-	0.19	0.10	0.09	0.81	-111.56	-48.2	-48.91
	✓	-	0.57	0.50	0.56	0.43	-87.50	-43.19	-43.70
HiER	-	✓	0.44	0.38	0.38	0.56	-98.72	-40.96	-42.28
	✓	✓	**0.75**	**0.80**	**0.83**	**0.25**	**-73.14**	**-31.35**	**-32.48**

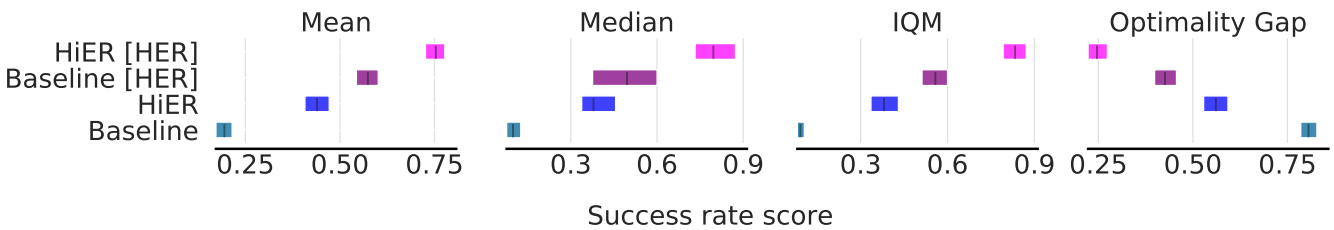


FIGURE 3: HiER compared to the state-of-the-art across all tasks with 95% CIs. Both HiER version outperform their corresponding baseline. HiER [HER] yields the best performance in all metrics. The point estimates are presented in Tab. 1

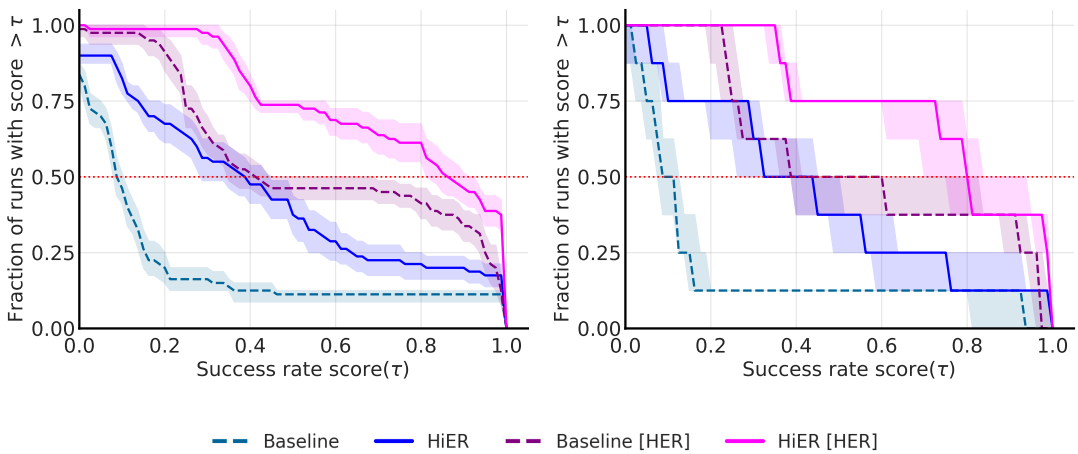


FIGURE 4: Performance profiles across all tasks with 95% CIs. **Left**: run-score distribution, **right**: average-score distribution. The red-dotted line shows the median values while the areas under the performance profiles correspond to the mean values (comparing with Tab. 1, the average-score distribution needs to be examined). Both HiER and HiER [HER] have stochastic dominance over their corresponding baselines.

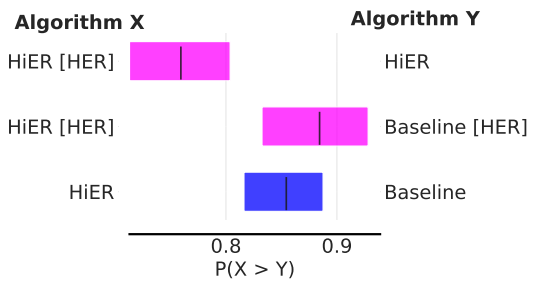


FIGURE 5: Probability of improvement of HiER versions compared to their corresponding baselines and themselves across all tasks with 95% CIs. The average probabilities from top to bottom are the following: 0.76, 0.88, and 0.85.

tasks are depicted in Fig. 6. For further details, we refer the reader to [37].

The aggregated results are presented in Fig. 7, while the performance profiles of the algorithms are demonstrated in the left side of Fig. 8. Our experimental results show that HiER (blue) and both versions of HiER+ (purple and magenta) significantly outperform the baselines (gray), while E2H-ISE alone could only slightly improve the performance. Moreover, the right side of Fig. 8 shows at least a 0.99 average probability of improvement for our methods compared to the baselines.

Regarding the specific tasks, the learning curves of the selected configurations are depicted in Fig. 6. For all cases, HiER and HiER+ significantly outperform the baselines. Moreover, Tab. 2 presents a simplified summary of the performance of the algorithms on the specific tasks. Our results show that HiER [HER] enhances its baseline by an increment of 0.03, 0.44, and 0.12 IQM score on the `PandaPush-v3`, `PandaSlide-v3`, and `PandaPickAndPlace-v3` tasks. Nevertheless, HiER+ [HER] further improves the performance, achieving 1.0, 0.82, and 0.71 IQM scores. Tab. 3 and Tab. 4 display the results of all configurations based on their success rates and rewards.

D. GYMNASIUM-ROBOTICS FETCH

In this section, HiER is evaluated on the `FetchPush-v2`, `FetchSlide-v2`, and `FetchPickAndPlace-v2` tasks of the MuJoCo-based Gymnasium-Robotics Fetch environment.

Even though the tasks are similar to the Panda-Gym robotic benchmark, the robot configuration, the observation space, and the environment dynamic (different simulator) are disparate. Our goal with these experiments is to demonstrate that HiER does not uniquely work for the Panda-Gym robotic benchmark. The tasks are depicted in Fig. 9. For more details, we refer the reader to [38].

In this section, HiER and HiER [HER] are compared with their corresponding baselines. Our experiment results are presented in Tab. 5 and depicted in Fig. 9 and Fig. 10. In all cases, the HiER versions outperform their corresponding baselines. Regarding the `FetchPush-v2` task, HiER [HER] improves the IQM score of the Baseline [HER] method by 0.06 (increasing from 0.92 to 0.98). In the case of the `FetchSlide-v2` task, HiER achieves the best result with a 0.56 IQM score, yielding a 0.54 increase compared to its baseline with 0.02. Interestingly, adding HER worsens the performance. Nevertheless, HiER [HER] still outperforms Baseline [HER]. Finally, for the `FetchPickAndPlace-v2` task, HiER [HER] achieves a 0.73 IQM score. Compared to the Baseline [HER] method with 0.24, it yields a 0.49 improvement. Interesting to note that for the latter two tasks, both HiER versions outperform both baselines.

E. GYMNASIUM-ROBOTICS POINTMAZE

In this section, HiER is evaluated on the `PointMaze-Wall-v3` and `PointMaze-S-v3` tasks of the MuJoCo-based Gymnasium-Robotics PointMaze environment to show the universality of our approach in a fundamentally different problem.

In these tasks, a ball, placed in a maze, needs to move from the start position to the goal position in a continuous state and action space. The start and the target positions are generated randomly with some constraints. For more details, we refer the reader to [39].

In our experiments, two different maze layouts were considered as depicted in Fig. 11. The reward function is changed to Eq. (1). As the tasks take longer to execute, the horizon is 500 timestep which is tenfold compared to the robotic manipulation tasks. Thus, for these experiments, the discount factor γ was set to one[22].

The results of our experiments are presented in Tab. 6 and depicted in Fig. 12. In the case of the `PointMaze-Wall-v3` task, the results are quite close to the optimal 1.0 success rate, thus there is no significant difference, even though HiER still performs equally or better than the baselines depending on the metrics and the configurations. Regarding the more challenging `PointMaze-S-v3` task, HiER [HER] outperforms Baseline [HER] by 0.2 IQM score, rising from 0.69 to 0.89.

F. QUALITATIVE EVALUATION

In this section, the qualitative evaluation of the aforementioned tasks is presented. We refer the reader to the project site[23] to watch our results compared with the baselines.

Regarding the Panda-Gym and the Gymnasium-Robotics Fetch environment, on many occasions, the baseline appears to be disoriented and incapable of completing the task. It appears that, during the training process, the agent became entrapped in a local minimum as a result of the challenging exploration problem caused by the continuous state and action space, the sparse reward, and the lack of demonstrations. This phenomenon is significantly less frequent in the case of HiER and HiER+ which solve the tasks with a considerably higher success rate, in correlation with the presented quantitative evaluation.

In the case of the Gymnasium-Robotics PointMaze environment, the qualitative evaluation does not show relevant differences. The primary reason is that while the mean, median, and IQM success rate score is considerably higher in the case of HiER [HER], both HiER [HER] and Baseline [HER] managed to obtain a perfect success rate of 100% at least once in the PointMaze environment (see Tab. 6).

G. OTHER

In this section, the different λ, ξ, and c methods are presented in Section V-G1 and V-G2. Additionally, our method is validated with DDPG and TD3 in Section V-G3. All experiments were conducted on the Panda-Gym benchmark.

[22]Not having a discount on future reward does not pose a problem as the reward function is formulated with -1 reward in every timestep, described in Eq.(1). Thus, the agent aims to solve the task as fast as possible.

[23]http://www.danielhorvath.eu/hier/#bookmark-qualitative-eval

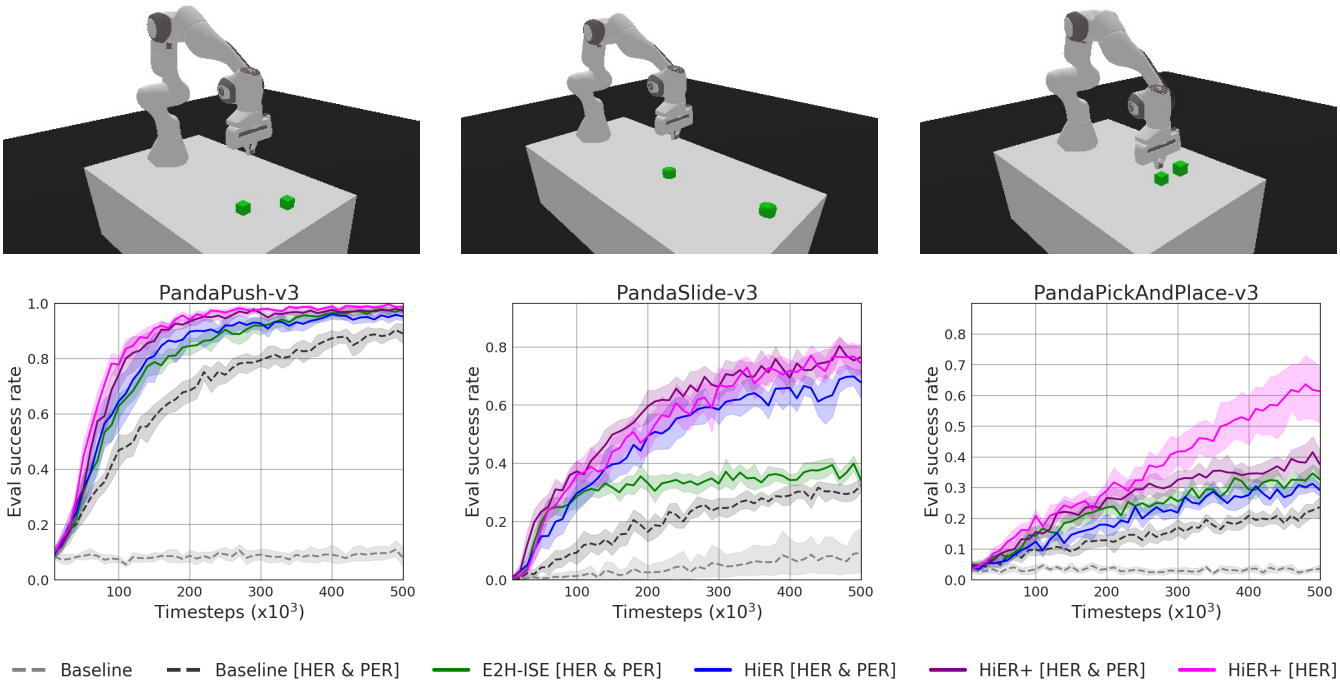


FIGURE 6: Learning curves of HiER and HiER+ with E2H-ISE compared to the state-of-the-art based on success rates on the push, slide, and pick-and-place tasks of the Panda-Gym robotic benchmark with 95% CIs.

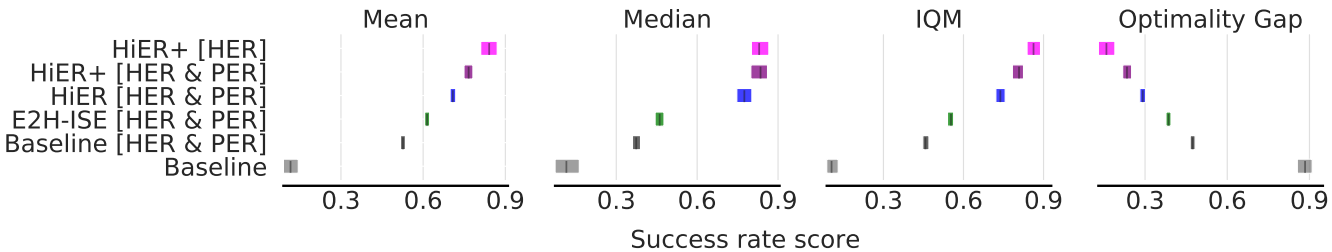


FIGURE 7: Aggregate metrics on the push, slide, and pick-and-place tasks of the Panda-Gym robotic benchmark with 95% CIs. HiER (blue) and both versions of HiER+ (purple and magenta) significantly outperform the baselines (gray). E2H-ISE alone could slightly improve the performance of the baseline.

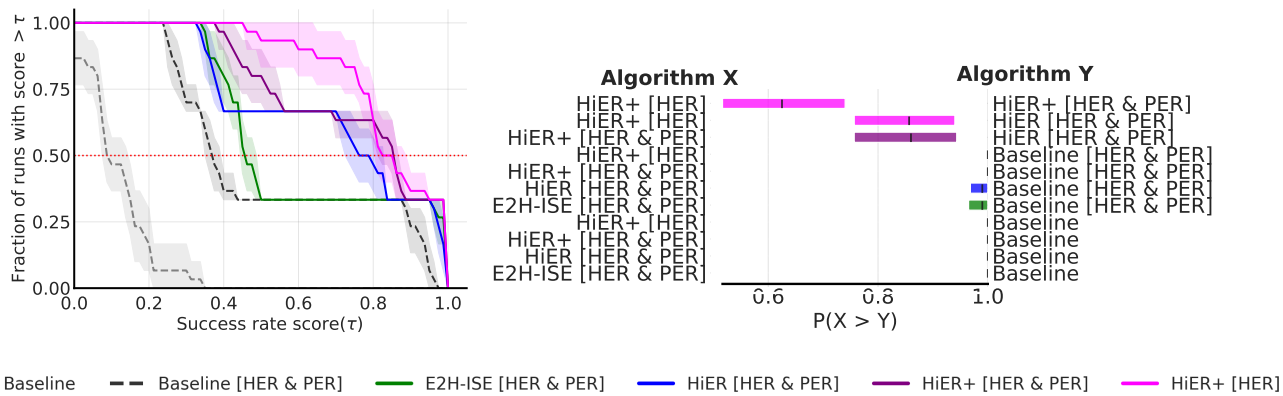


FIGURE 8: **Left:** performance profiles (run-score distribution) on the push, slide, and pick-and-place tasks of the Panda-Gym robotic benchmark with 95% CIs. **Right:** Probability of improvement on the push, slide, and pick-and-place tasks of the Panda-Gym robotic benchmark with 95% CIs. The average probabilities from top to bottom: 0.625, 0.857, 0.86, 1.0, 1.0, 0.99, 0.99, 1.0, 1.0, and 1.0.

TABLE 2: Simplified summary of our results on the push, slide, and pick-and-place tasks of the Panda-Gym robotic benchmark based on success rates. The column-wise best results are marked in bold. The full table with all the configurations is presented in Tab. 3.

	PandaPush-v3 | PandaSlide-v3 | PandaPickAndPlace-v3					
	Mean ↑	Median ↑	IQM ↑	OG ↓	Max ↑	Std ↓
Baseline [HER]	0.97 | 0.38 | 0.27	0.98 | 0.37 | 0.28	0.97 | 0.37 | 0.27	0.03 | 0.62 | 0.73	0.99 | 0.45 | 0.32	0.02 | **0.04** | 0.03
HiER [HER]	**1.00** | 0.79 | 0.39	**1.00** | **0.81** | 0.39	**1.00** | 0.81 | 0.39	**0.00** | 0.21 | 0.61	**1.00** | 0.91 | 0.42	**0.00** | 0.09 | **0.02**
HiER+ [HER]	**1.00** | **0.83** | **0.69**	**1.00** | **0.81** | **0.74**	**1.00** | **0.82** | **0.71**	**0.00** | **0.17** | **0.31**	**1.00** | **0.95** | **0.90**	**0.00** | 0.05 | 0.14

TABLE 3: HiER and HiER+ compared to the state-of-the-art based on success rates on the Panda-Gym robotic benchmark. On the left side of the header, the components of the specific algorithm are displayed (HER, PER, ISE, HiER). The column-wise best results are marked in bold.

	HER PER ISE HiER	PandaPush-v3 | PandaSlide-v3 | PandaPickAndPlace-v3					
		Mean ↑	Median ↑	IQM ↑	OG ↓	Max ↑	Std ↓
Baselines	- - - -	0.16 | 0.12 | 0.07	0.15 | 0.05 | 0.07	0.15 | 0.08 | 0.07	0.84 | 0.88 | 0.93	0.21 | 0.34 | 0.09	0.03 | 0.13 | **0.01**
	✓- - -	0.97 | 0.38 | 0.27	0.98 | 0.37 | 0.28	0.97 | 0.37 | 0.27	0.03 | 0.62 | 0.73	0.99 | 0.45 | 0.32	0.02 | 0.04 | 0.03
	- ✓- -	0.26 | 0.25 | 0.08	0.25 | 0.27 | 0.09	0.25 | 0.27 | 0.08	0.74 | 0.75 | 0.92	0.43 | 0.42 | 0.10	0.07 | 0.14 | **0.01**
	✓✓- -	0.93 | 0.37 | 0.28	0.94 | 0.37 | 0.28	0.93 | 0.37 | 0.27	0.07 | 0.63 | 0.72	0.97 | 0.43 | 0.33	0.03 | **0.02** | 0.02
HiER	- - - ✓	0.44 | 0.29 | 0.09	0.44 | 0.28 | 0.09	0.44 | 0.29 | 0.09	0.56 | 0.71 | 0.91	0.57 | 0.39 | 0.11	0.09 | 0.07 | **0.01**
	✓- - ✓	**1.00** | 0.79 | 0.39	**1.00** | 0.81 | 0.39	**1.00** | 0.81 | 0.39	**0.00** | 0.21 | 0.61	**1.00** | 0.91 | 0.42	**0.00** | 0.09 | 0.02
	- ✓- ✓	0.80 | 0.41 | 0.13	0.88 | 0.42 | 0.13	0.83 | 0.44 | 0.13	0.20 | 0.59 | 0.87	0.98 | 0.66 | 0.16	0.15 | 0.17 | 0.02
	✓✓- ✓	0.98 | 0.78 | 0.37	0.99 | 0.78 | 0.37	0.99 | 0.78 | 0.37	0.02 | 0.22 | 0.63	**1.00** | 0.83 | 0.39	0.01 | 0.05 | 0.02
ISE	- - ✓-	0.85 | 0.45 | 0.25	0.85 | 0.45 | 0.25	0.86 | 0.45 | 0.25	0.15 | 0.55 | 0.75	0.95 | 0.47 | 0.30	0.06 | **0.02** | 0.03
	✓- ✓-	**1.00** | 0.45 | 0.42	**1.00** | 0.45 | 0.43	**1.00** | 0.45 | 0.43	**0.00** | 0.55 | 0.58	**1.00** | 0.52 | 0.53	0.01 | 0.03 | 0.04
	- ✓✓-	0.83 | 0.44 | 0.31	0.83 | 0.44 | 0.30	0.83 | 0.44 | 0.30	0.17 | 0.56 | 0.69	0.89 | 0.52 | 0.36	0.04 | 0.04 | 0.03
	✓✓✓-	0.99 | 0.46 | 0.39	**1.00** | 0.46 | 0.38	**1.00** | 0.46 | 0.39	0.01 | 0.54 | 0.61	**1.00** | 0.50 | 0.44	0.01 | 0.03 | 0.03
HiER+	- - ✓✓	0.98 | 0.53 | 0.33	0.99 | 0.48 | 0.32	0.99 | 0.49 | 0.32	0.02 | 0.47 | 0.67	**1.00** | 0.76 | 0.39	0.01 | 0.12 | 0.03
	✓- ✓✓	**1.00** | 0.83 | **0.69**	**1.00** | 0.81 | **0.74**	**1.00** | 0.82 | **0.71**	**0.00** | 0.17 | **0.31**	**1.00** | **0.95** | **0.90**	**0.00** | 0.05 | 0.14
	- ✓✓✓	0.98 | 0.51 | 0.41	0.98 | 0.49 | 0.40	0.98 | 0.50 | 0.40	0.02 | 0.49 | 0.59	**1.00** | 0.65 | 0.50	0.02 | 0.07 | 0.05
	✓✓✓✓	**1.00** | **0.84** | 0.47	**1.00** | **0.86** | 0.45	**1.00** | **0.85** | 0.46	**0.00** | **0.16** | 0.53	**1.00** | 0.88 | 0.55	**0.00** | 0.05 | 0.06

TABLE 4: HiER and HiER+ compared to the state-of-the-art based on the evaluation rewards on the Panda-Gym robotic benchmark. On the left side of the header, the components of the specific algorithm are displayed (HER, PER, ISE, HiER). The desired performance scores for the OG metric are -10, -20, and -30 for the push, slide, and pick-and-place tasks respectively. The column-wise best results are marked in bold.

	HER PER ISE HiER	PandaPush-v3 | PandaSlide-v3 | PandaPickAndPlace-v3					
		Mean ↑	Median ↑	IQM ↑	OG ↓	Max ↑	Std ↓
Baselines	- - - -	-46.2 | -46.7 | -48.2	-46.2 | -49.0 | -48.5	-46.4 | -48.0 | -48.3	36.2 | 26.7 | 18.2	-41.0 | -37.6 | -46.5	2.4 | 4.3 | 1.0
	✓- - -	-11.4 | -39.2 | -41.8	-11.2 | -38.5 | -41.1	-11.2 | -38.7 | -41.5	1.5 | 19.2 | 11.8	-9.6 | -36.5 | -39.5	1.3 | 2.0 | 1.9
	- ✓- -	-40.8 | -43.5 | -48.8	-41.6 | -43.4 | -48.5	-41.4 | -43.0 | -48.7	30.8 | 23.5 | 18.8	-32.6 | -38.3 | -48.0	3.5 | 3.8 | **0.6**
	✓✓- -	-12.7 | -38.5 | -40.4	-11.9 | -38.7 | -39.8	-12.3 | -38.6 | -40.1	2.7 | 18.5 | 10.4	-9.7 | -35.5 | -36.6	2.5 | **1.5** | 2.5
HiER	- - - ✓	-34.1 | -42.0 | -47.6	-35.0 | -42.0 | -47.8	-34.4 | -42.5 | -47.7	24.1 | 22.0 | 17.6	-27.0 | -35.6 | -46.2	4.5 | 3.5 | 0.8
	✓- - ✓	-7.0 | -23.6 | -37.2	**-6.8** | -22.6 | -36.6	**-6.9** | -23.1 | -36.9	**0.0** | 4.0 | 7.2	-6.1 | -17.8 | -34.5	0.7 | 4.1 | 1.9
	- ✓- ✓	-17.2 | -38.2 | -46.8	-14.5 | -36.6 | -46.7	-15.7 | -36.8 | -46.7	7.2 | 18.2 | 16.8	-9.9 | -32.9 | -45.1	7.6 | 5.1 | 1.2
	✓✓- ✓	-8.4 | -25.4 | -37.7	-8.3 | -24.9 | -37.3	-8.3 | -25.1 | -37.5	0.1 | 5.4 | 7.7	-6.4 | -21.2 | -36.2	1.2 | 3.0 | 1.4
ISE	- - ✓-	-14.9 | -37.3 | -41.6	-15.0 | -37.8 | -41.6	-15.1 | -37.4 | -41.4	5.0 | 17.3 | 11.6	-8.3 | -33.6 | -38.8	3.4 | 2.2 | 2.0
	✓- ✓-	-8.1 | -35.9 | -34.1	-8.0 | -35.9 | -34.6	-8.0 | -35.9 | -34.4	**0.0** | 15.9 | 4.3	-6.7 | -32.7 | -27.2	1.0 | 2.0 | 3.2
	- ✓✓-	-16.3 | -37.3 | -39.1	-15.7 | -37.5 | -39.0	-16.1 | -37.5 | -38.9	6.3 | 17.3 | 9.1	-12.1 | -34.8 | -35.8	3.2 | **1.5** | 2.0
	✓✓✓-	-8.1 | -37.6 | -36.1	-7.8 | -37.7 | -36.6	-8.0 | -37.7 | -36.5	**0.0** | 17.6 | 6.1	-6.4 | -34.9 | -31.6	1.2 | **1.5** | 2.1
HiER+	- - ✓✓	-8.8 | -33.4 | -38.1	-8.2 | -33.1 | -38.5	-8.5 | -33.8 | -38.3	0.3 | 13.4 | 8.1	-7.1 | -26.6 | -34.8	1.7 | 3.8 | 1.7
	✓- ✓✓	**-6.9** | -22.5 | **-24.2**	-7.0 | -22.9 | **-23.2**	-7.0 | -22.8 | **-23.0**	**0.0** | 3.0 | **1.0**	**-5.9** | **-17.4** | **-15.0**	**0.6** | 3.2 | 6.3
	- ✓✓✓	-8.6 | -34.9 | -34.6	-8.4 | -35.9 | -34.8	-8.5 | -35.3 | -34.6	**0.0** | 14.9 | 4.6	-7.8 | -30.5 | -30.9	0.7 | 2.6 | 1.9
	✓✓✓✓	-7.7 | **-21.7** | -33.9	-7.4 | **-21.5** | -34.4	-7.5 | **-21.5** | -34.0	**0.0** | **2.3** | 4.1	-7.0 | **-17.4** | -28.0	**0.6** | 3.1 | 3.3

TABLE 5: HiER compared to the state-of-the-art based on success rates on push, slide, and pick-and-place tasks of the Gymnasium-Robotics Fetch benchmark. The column-wise best results are marked in bold.

	HER HiER	FetchPush-v2 | FetchSlide-v2 | FetchPickAndPlace-v2					
		Mean ↑	Median ↑	IQM ↑	OG ↓	Max ↑	Std ↓
Baselines	- -	0.12 | 0.02 | 0.08	0.12 | 0.02 | 0.08	0.12 | 0.02 | 0.08	0.88 | 0.98 | 0.92	0.14 | 0.04 | 0.10	**0.01** | **0.01** | **0.01**
	✓-	0.92 | 0.23 | 0.24	0.93 | 0.22 | 0.23	0.92 | 0.22 | 0.24	0.08 | 0.77 | 0.76	0.98 | 0.39 | 0.30	0.05 | 0.07 | 0.04
HiER	- ✓	0.76 | **0.56** | 0.32	0.93 | **0.55** | 0.17	0.83 | **0.56** | 0.26	0.24 | **0.44** | 0.68	**1.00** | **0.80** | 0.76	0.29 | 0.13 | 0.22
	✓✓	**0.98** | 0.35 | **0.73**	**0.99** | 0.36 | **0.77**	**0.98** | 0.36 | **0.73**	**0.02** | 0.65 | **0.27**	**1.00** | 0.39 | **0.93**	0.02 | 0.03 | 0.14

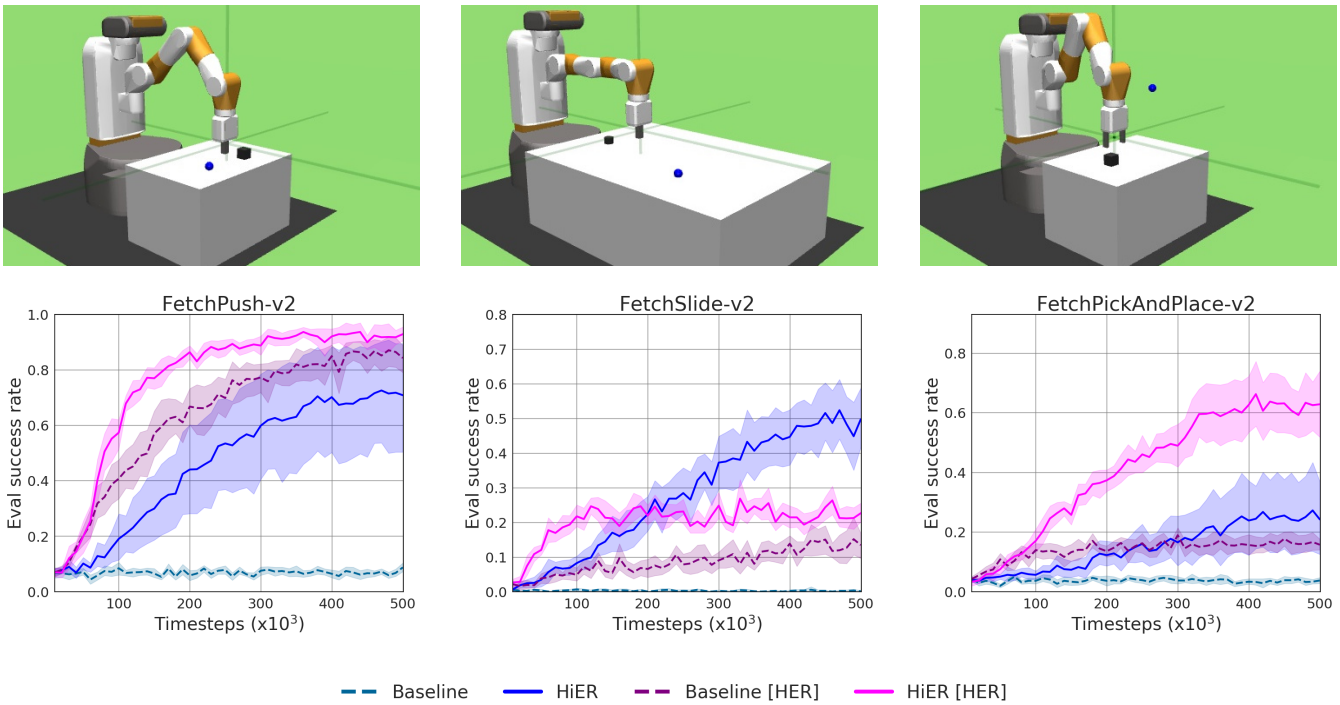


FIGURE 9: Learning curves of HiER compared with its baselines on push, slide, and pick-and-place tasks of the Gymnasium-Robotics Fetch benchmark with 95% CIs.

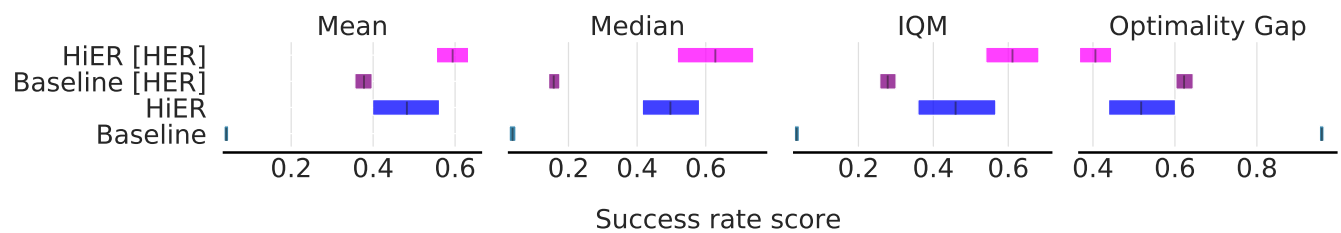
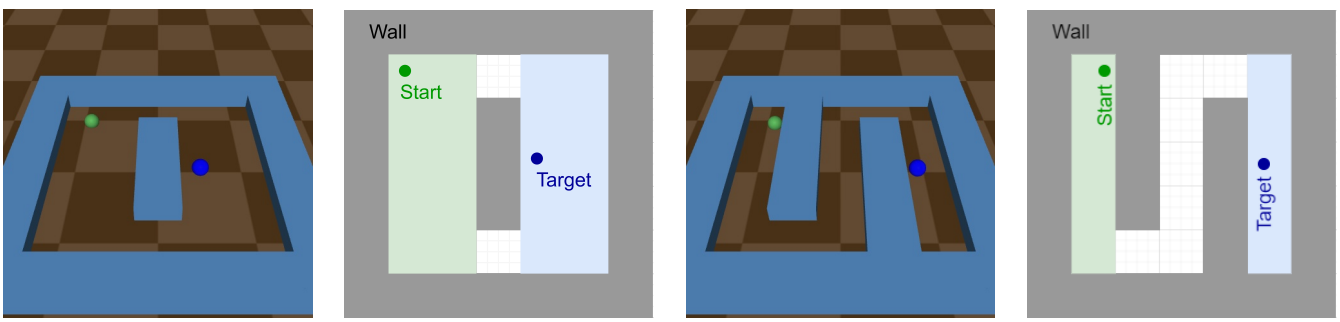


FIGURE 10: Aggregate metrics on the push, slide, and pick-and-place tasks of the Gymnasium-Robotics Fetch benchmark with 95% CIs. Both HiER (blue) and HiER [HER] (magenta) significantly outperform the baselines (light blue and purple).



(a) `PointMaze-Wall-v3` **(b)** Layout of `PointMaze-Wall-v3` **(c)** `PointMaze-S-v3` **(d)** Layout of `PointMaze-S-v3`

FIGURE 11: The tasks of Gymnasium-Robotics PointMaze environment [39]. The mazes were custom-made, thus we named them accordingly. The layouts (b) and (d) show the placement of the walls and the possible start and target positions from a top view. The environment is based on the MuJoCo simulator [41].

1) HiER λ and ξ methods

The comparison of the different HiER λ methods are depicted in Fig. 13 (a) and (b) and Fig. 14. The experiments were executed without HER, PER, and E2H-ISE. In these settings, the `predefined` λ method outperforms the other variants, although its CI overlaps the CI of the `fix` λ method. The λ profiles are presented in Fig. 13 (b).

The impact of HiER ξ method is shown in Fig. 13 (c) and Fig. 15. The experiments were executed with HER and E2H-ISE but without PER. In these settings, the `fix` $\xi = 0.25$, $\xi = 0.5$, and the `prioritized` ξ method appear to be the best versions in this order, although their CIs overlap[24]. Important to note, that when PER is active, it scales the gradient proportionally to the probability of the samples, thus `prioritized` ξ mode is recommended to counterbalance this effect.

2) E2H-ISE versions

The different E2H-ISE c methods are presented in Tab. 7 and displayed in Fig. 16. The experiments were executed without PER. The ranking of E2H-ISE versions is relatively sensible for the applied methods (HER and HiER). Without HiER, there is no significant difference between the c methods. With HiER but without HER the `control` and the `control adaptive` c methods yield the highest performance, although their CIs overlap with the other versions. With HiER and HER, the `control adaptive` and `self-paced` c methods achieve the best performance. Nevertheless, further optimization, or possibly another version of E2H-ISE could improve the performance.

3) TD3 and DDPG

To validate our methods not only with SAC, Fig. 17 and Tab.8 show our results in the case of DDPG and TD3. In both cases, HiER+ improved the results of the baseline. In the case of TD3 (blue), the improvement is more significant as the CIs do not overlap. In the case of DDPG (magenta), although there is a considerable improvement, the CIs overlap. Note that DDPG is less stable than TD3 resulting in wider CIs.

VI. CONCLUSION

In this work, we introduced a novel technique called the highlight experience replay (HiER) to facilitate the training of off-policy reinforcement learning agents in a robotic, sparse-reward environment with continuous state and action spaces. Furthermore, the agent is devoid of access to any form of demonstration. These constraints, significantly exacerbate the difficulty of exploration.

In our method, a secondary replay buffer is created to store the most relevant experiences based on some criteria. At training, the transitions are sampled from both the standard experience replay buffer and the highlight experience replay buffer. Similarly to the hindsight experience replay (HER) and prioritized experience replay (PER), HiER can be added to any off-policy reinforcement learning algorithm. Following [34], HiER is classified as a data exploitation (or implicit) curriculum learning method.

To demonstrate the universality of HiER, it was validated on 8 tasks of three different robotics benchmarks [37]–[39] based on two different simulators [40], [41]. On one hand, among the 8 tasks, 3-3 were the same in principle (push, slide, and pick-and-place) but the robot configurations, the state spaces, and the dynamics of the environments were disparate. On the other hand, the last 2 tasks were fundamentally different as a ball needed to find a target in different mazes.

In all of the experiments, HiER significantly improved the state-of-the-art methods. Our experimental results show that HiER is especially beneficial in hard-to-solve tasks such as `PandaSlide-v3`, `FetchPickAndPlace-v2`, or `PointMaze--S-v3`.

HiER collects and stores positive experiences to improve the training process. With HiER+, we showed how HiER can benefit from a traditional, data collection curriculum learning method as well. Lack of general and easy-to-implement solutions, we proposed E2H-ISE, an *easy2hard* data collection CL method that requires minimal prior knowledge and controls the entropy of the initial state-goal distribution $\mathcal{H}(\mu_0)$ which indirectly controls the task difficulty. Nevertheless, applying more sophisticated CL methods in place of E2H-ISE might be beneficial in future research.

HiER+ was validated on the `PandaPush-v3`, `PandaSlide-v3`, and `PandaPickAndPlace-v3` tasks of the Panda-Gym [37] robotic benchmark. Our results show that HiER+ could further improve the performance of HiER.

Furthermore, we presented our experiments on the different λ, ξ, and c methods of HiER and E2H-ISE. On one hand, we found that in the case of HiER λ, the `predefined` version was superior. On the other hand, the rankings of the ξ and c methods are more unambiguous and depend on the applied configuration. We also showed that HiER+ improves the baselines not only with SAC but with TD3 and DDPG as well.

Additionally, the qualitative analysis revealed that HiER and HiER+ showed a reduced tendency to be trapped in local minima compared to the vanilla baseline methods.

For future work, we will investigate other possible HiER versions. Moreover, we are interested in how HiER+ could facilitate sim2sim and sim2real knowledge transfer.

[24]In other settings, we found $\xi = 0.5$ slightly better than the others.

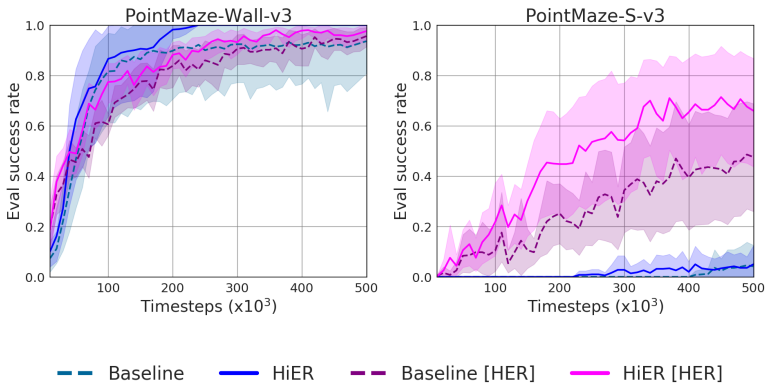
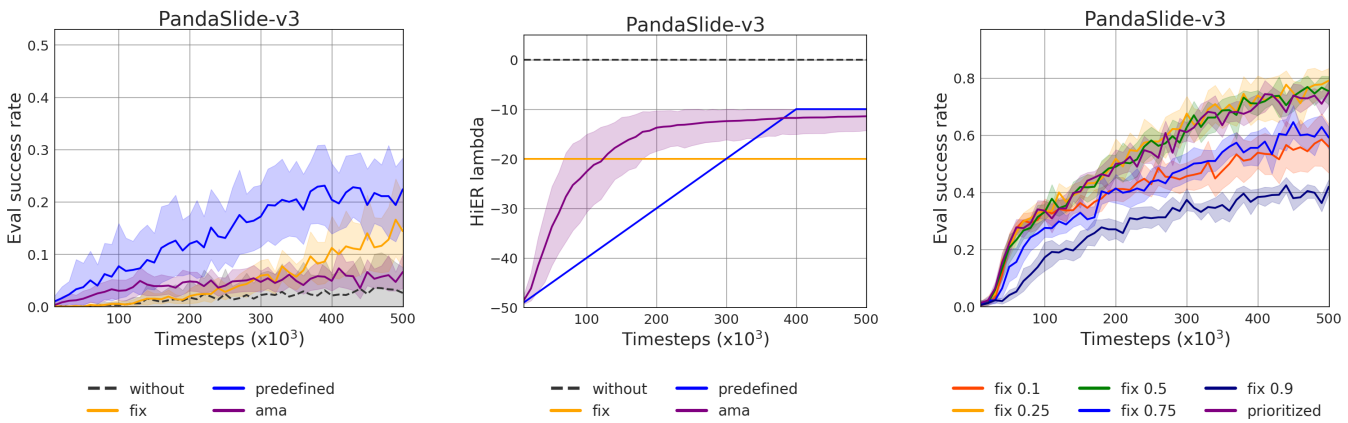


FIGURE 12: Learning curves of HiER compared with its baselines on the Gymnasium-Robotics PointMaze environment with 95% CIs.

TABLE 6: HiER compared to the state-of-the-art based on success rates on the Gymnasium-Robotics PointMaze environment. The column-wise best results are marked in bold.

	HER HiER	PointMaze-Wall-v3 | PointMaze-S-v3					
		Mean ↑	Median ↑	IQM ↑	OG ↓	Max ↑	Std ↓
Baselines	- -	0.94 | 0.05	**1.00** | 0.00	**1.00** | 0.00	0.06 | 0.95	**1.00** | 0.46	0.19 | 0.14
	✓-	0.97 | 0.61	**1.00** | 0.76	0.99 | 0.69	0.03 | 0.39	**1.00** | **1.00**	0.05 | 0.36
HiER	- ✓	**1.00** | 0.05	**1.00** | 0.00	**1.00** | 0.00	**0.00** | 0.95	**1.00** | 0.28	**0.00** | **0.11**
	✓✓	**1.00** | **0.80**	**1.00** | **0.91**	**1.00** | **0.89**	**0.00** | **0.20**	**1.00** | **1.00**	0.01 | 0.29



(a) The effect of HiER λ versions on the success rate. ξ is fixed at 0.5.

(b) The change of λ values over time. ξ is fixed at 0.5.

(c) The effect of HiER ξ methods. HiER λ is set to `predefined`.

FIGURE 13: The analysis of HiER λ versions (a) and (b), and HiER ξ versions (c). HiER λ `ama` parameters: $\lambda_0 = -50$, $\lambda_{max} = -10$ $M = 0$ and $w = 20$. The *without* version indicates that HiER was not used.

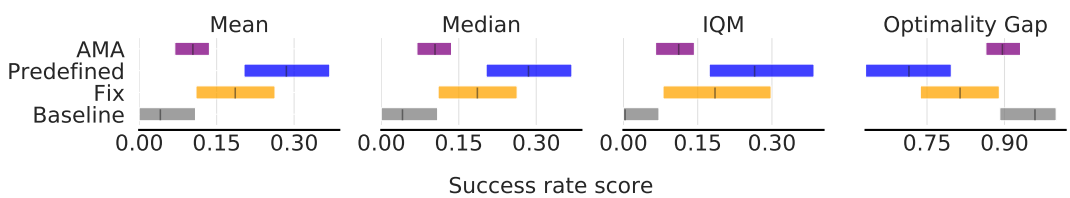


FIGURE 14: Comparison of different HiER λ methods on the slide task of the Panda-Gym benchmark with 95% CIs. The `predefined` λ method is seemingly superior, although the CIs with the `fix` λ method overlap. HiER λ `ama` parameters: $\lambda_0 = -50$, $\lambda_{max} = -10$ $M = 0$ and $w = 20$. The profiles of HiER λ are depicted on Fig. 13 (b).

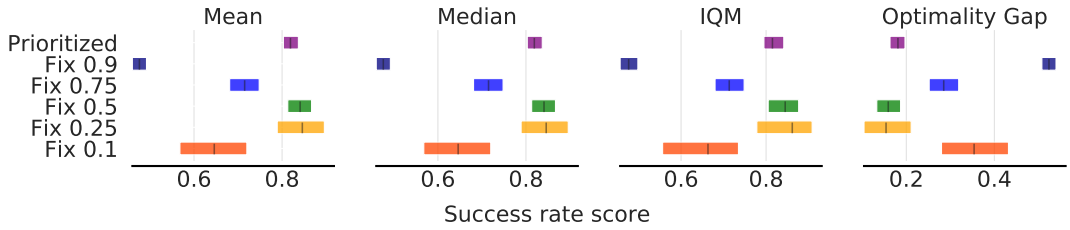


FIGURE 15: Comparison of different HiER ξ methods on the slide task of the Panda-Gym benchmark with 95% CIs. The `fix` $\xi = 0.25$, $\xi = 0.5$, and the `prioritized` appear to be the best versions in this order, although their CIs overlap.

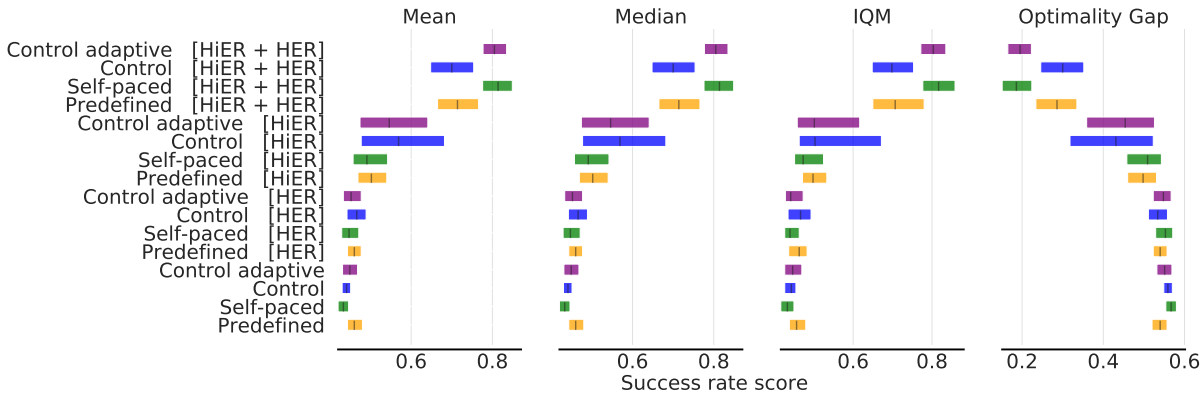


FIGURE 16: Comparison of different E2H-ISE c methods on the slide task of the Panda-Gym benchmark with 95% CIs. The parameters of the methods and the point estimates are presented in Tab. 7.

TABLE 7: The effect of the E2H-ISE c methods on the success rates on the `PandaSlide-v3` task. HiER parameters: λ mode `predefined` and ξ `fix` with $\xi = 0.5$. E2H-ISE parameters: `self-paced` $\Psi_{low} = 0.2$, $\Psi_{high} = 0.8$ and $\delta = 0.05$; `control`: $\psi = 0.8$ and $\delta = 0.01$; `control adaptive`: $\Delta = 0.2$, $\psi_{max} = 0.9$, and $\delta = 0.01$. The row-wise best results are marked in bold.

Components		`predefined`			`self-paced`			`control`			`control adaptive`		
HER	HiER	Max ↑	Mean ↑	Std ↓	Max ↑	Mean ↑	Std ↓	Max ↑	Mean ↑	Std ↓	Max ↑	Mean ↑	Std ↓
-	-	**0.52**	**0.46**	0.03	0.46	0.43	0.02	0.46	0.44	0.02	0.51	0.45	0.03
✓	-	0.49	**0.46**	0.03	**0.54**	0.45	0.03	0.53	0.47	0.04	**0.54**	0.45	0.04
-	✓	0.63	0.50	0.06	0.69	0.49	0.07	**0.95**	**0.57**	0.17	0.90	0.55	0.14
✓	✓	0.85	0.71	0.08	**0.90**	**0.81**	0.06	0.87	0.70	0.08	**0.90**	0.80	0.05

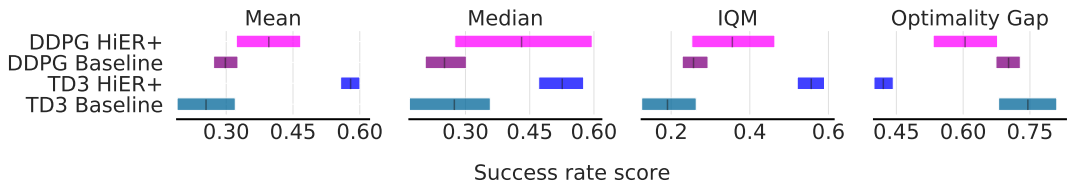


FIGURE 17: Comparison of the TD3 and DDPG versions of HiER+ wi with their baselines on the push, slide, and pick-and-place tasks of the Panda-Gym benchmark with 95% CIs. The point estimates are presented in Tab. 8.

TABLE 8: HiER+ compared to the state-of-the-art based on success rates on the Panda-Gym robotic benchmark in the case of TD3 and DDPG. The column-wise best results for TD3 and DDPG separately are marked in bold.

		`PandaPush-v3` | `PandaSlide-v3` | `PandaPickAndPlace-v3`					
	RL Algorithm	Mean ↑	Median ↑	IQM ↑	OG ↓	Max ↑	Std ↓
DDPG	Baseline	0.25 | 0.56 | 0.08	0.23 | 0.56 | 0.08	0.24 | 0.56 | 0.08	0.75 | 0.44 | 0.92	0.42 | 0.74 | 0.11	**0.08** | **0.11** | **0.01**
	HiER+	**0.43** | **0.63** | **0.13**	**0.32** | **0.68** | **0.12**	**0.38** | **0.68** | **0.12**	**0.57** | **0.37** | **0.87**	**0.91** | **0.83** | **0.20**	0.27 | 0.22 | 0.03
TD3	Baseline	0.40 | 0.27 | 0.09	0.28 | 0.36 | 0.09	0.36 | 0.30 | 0.09	0.60 | 0.73 | 0.91	0.86 | 0.44 | 0.11	0.28 | 0.16 | **0.01**
	HiER+	**0.93** | **0.53** | **0.28**	**0.94** | **0.56** | **0.30**	**0.94** | **0.54** | **0.29**	**0.07** | **0.47** | **0.72**	**0.99** | **0.63** | **0.35**	**0.04** | **0.08** | 0.05

REFERENCES

[1] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345–1359, Oct. 2010, conference Name: IEEE Transactions on Knowledge and Data Engineering. [Online]. Available: http://doi.org/10.1109/TKDE.2009.191

[2] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A Survey of Transfer Learning," Journal of Big Data, vol. 3, no. 1, p. 9, Dec. 2016. [Online]. Available: http://doi.org/10.1186/s40537-016-0043-6

[3] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino, "Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning," IEEE Access, vol. 9, pp. 153 171–153 187, 2021. [Online]. Available: http://doi.org/10.1109/ACCESS.2021.3126658

[4] A. Barisic, F. Petric, and S. Bogdan, "Sim2Air - Synthetic Aerial Dataset for UAV Monitoring," IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 3757–3764, Apr. 2022, conference Name: IEEE Robotics and Automation Letters. [Online]. Available: http://doi.org/10.1109/LRA.2022.3147337

[5] D. Horváth, G. Erdős, Z. Istenes, T. Horváth, and S. Földi, "Object Detection Using Sim2Real Domain Randomization for Robotic Applications," IEEE Transactions on Robotics, vol. 39, pp. 1225–1243, Apr. 2023. [Online]. Available: http://doi.org/10.1109/TRO.2022.3207619

[6] D. Horváth, K. Bocsi, G. Erdős, and Z. Istenes, "Sim2Real Grasp Pose Estimation for Adaptive Robotic Applications," IFAC-PapersOnLine, vol. 56, no. 2, pp. 5233–5239, Jan. 2023. [Online]. Available: http://doi.org/10.1016/j.ifacol.2023.10.121

[7] S. Zhou, M. K. Helwa, A. P. Schoellig, A. Sarabakha, and E. Kayacan, "Knowledge Transfer Between Robots with Similar Dynamics for High-Accuracy Impromptu Trajectory Tracking," in 2019 18th European Control Conference (ECC), Jun. 2019, pp. 1–8. [Online]. Available: http://doi.org/10.23919/ECC.2019.8796140

[8] Y. Bao, Y. Li, S.-L. Huang, L. Zhang, L. Zheng, A. Zamir, and L. Guibas, "An Information-Theoretic Approach to Transferability in Task Transfer Learning," in 2019 IEEE International Conference on Image Processing (ICIP), Sep. 2019, pp. 2309–2313, iSSN: 2381-8549. [Online]. Available: http://doi.org/10.1109/ICIP.2019.8803726

[9] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: A Bradford Book, 2018.

[10] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A Gentle Introduction to Reinforcement Learning and its Application in Different Fields," IEEE Access, vol. 8, pp. 209 320–209 344, 2020. [Online]. Available: http://doi.org/10.1109/ACCESS.2020.3038605

[11] M. Q. Mohammed, K. L. Chung, and C. S. Chyi, "Review of Deep Reinforcement Learning-Based Object Grasping: Techniques, Open Challenges, and Recommendations," IEEE Access, vol. 8, pp. 178 450–178 481, 2020. [Online]. Available: http://doi.org/10.1109/ACCESS.2020.3027923

[12] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm," Dec. 2017, arXiv:1712.01815 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.1712.01815

[13] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the Game of Go without Human Knowledge," Nature, vol. 550, no. 7676, pp. 354–359, Oct. 2017, number: 7676 Publisher: Nature Publishing Group. [Online]. Available: https://doi.org/10.1038/nature24270

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-Level Control Through Deep Reinforcement Learning," Nature, vol. 518, no. 7540, pp. 529–533, Feb. 2015, number: 7540 Publisher: Nature Publishing Group. [Online]. Available: https://doi.org/10.1038/nature14236

[15] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in Proceedings of the 31st International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 1. Bejing, China: PMLR, 22–24 Jun 2014, pp. 387–395. [Online]. Available: https://proceedings.mlr.press/v32/silver14.html

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous Control with Deep Reinforcement Learning," Jul. 2019, arXiv:1509.02971 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.1509.02971

[17] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," Oct. 2018, arXiv:1802.09477 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.1802.09477

[18] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," Aug. 2018, arXiv:1801.01290 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.1801.01290

[19] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, "Active Domain Randomization," Jul. 2019, arXiv:1904.04762 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.1904.04762

[20] S. Luo, H. Kasaei, and L. Schomaker, "Accelerating Reinforcement Learning for Reaching Using Continuous Curriculum Learning," in 2020 International Joint Conference on Neural Networks (IJCNN), Jul. 2020, pp. 1–8, iSSN: 2161-4407. [Online]. Available: https://doi.org/10.1109/IJCNN48605.2020.9207427

[21] C. Florensa, D. Held, M. Wulfmeier, M. Zhang, and P. Abbeel, "Reverse Curriculum Generation for Reinforcement Learning," Jul. 2018, arXiv:1707.05300 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.1707.05300

[22] B. Ivanovic, J. Harrison, A. Sharma, M. Chen, and M. Pavone, "BaRC: Backward Reachability Curriculum for Robotic Reinforcement Learning," in 2019 International Conference on Robotics and Automation (ICRA), May 2019, pp. 15–21, iSSN: 2577-087X. [Online]. Available: https://doi.org/10.1109/ICRA.2019.8794206

[23] T. Salimans and R. Chen, "Learning Montezuma's Revenge from a Single Demonstration," Dec. 2018, arXiv:1812.03381 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.1812.03381

[24] S. Sukhbaatar, Z. Lin, I. Kostrikov, G. Synnaeve, A. Szlam, and R. Fergus, "Intrinsic Motivation and Automatic Curricula via Asymmetric Self-Play," Apr. 2018, arXiv:1703.05407 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.1703.05407

[25] C. Florensa, D. Held, X. Geng, and P. Abbeel, "Automatic Goal Generation for Reinforcement Learning Agents," Jul. 2018, arXiv:1705.06366 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.1705.06366

[26] V. H. Pong, M. Dalal, S. Lin, A. Nair, S. Bahl, and S. Levine, "Skew-Fit: State-Covering Self-Supervised Reinforcement Learning," Aug. 2020, arXiv:1903.03698 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.1903.03698

[27] S. Racaniere, A. K. Lampinen, A. Santoro, D. P. Reichert, V. Firoiu, and T. P. Lillicrap, "Automated Curricula Through Setter-Solver Interactions," Jan. 2020, arXiv:1909.12892 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.1909.12892

[28] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," Feb. 2016, arXiv:1511.05952 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.1511.05952

[29] J. Oh, Y. Guo, S. Singh, and H. Lee, "Self-Imitation Learning," Jun. 2018, arXiv:1806.05635 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.1806.05635

[30] J. Ferret, O. Pietquin, and M. Geist, "Self-Imitation Advantage Learning," Dec. 2020, arXiv:2012.11989 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.2012.11989

[31] C. Wang and K. Ross, "Boosting Soft Actor-Critic: Emphasizing Recent Experience without Forgetting the Past," Jun. 2019, arXiv:1906.04009 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.1906.04009

[32] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight Experience Replay," in Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1707.01495

[33] J. Bujalance and F. Moutarde, "Reward Relabelling for Combined Reinforcement and Imitation Learning on Sparse-Reward Tasks," in Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, 2023, pp. 2565–2567. [Online]. Available: https://doi.org/10.48550/arXiv.2201.03834

[34] R. Portelas, C. Colas, L. Weng, K. Hofmann, and P.-Y. Oudeyer, "Automatic Curriculum Learning For Deep RL: A Short Survey," May 2020, arXiv:2003.04664 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.2003.04664

[35] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in Proceedings of the 26th Annual International Conference on Machine Learning, ser. ICML '09. New York, NY, USA: Association

for Computing Machinery, Jun. 2009, pp. 41–48. [Online]. Available: https://doi.org/10.1145/1553374.1553380

[36] X. Wang, Y. Chen, and W. Zhu, "A Survey on Curriculum Learning," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 9, pp. 4555–4576, Sep. 2022, conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence. [Online]. Available: http://doi.org/10.1109/TPAMI.2021.3069908

[37] Q. Gallouédec, N. Cazin, E. Dellandréa, and L. Chen, "Panda-Gym: Open-Source Goal-Conditioned Environments for Robotic Learning," Dec. 2021, arXiv:2106.13687 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.2106.13687

[38] M. Plappert, M. Andrychowicz, A. Ray, B. McGrew, B. Baker, G. Powell, J. Schneider, J. Tobin, M. Chociej, P. Welinder, V. Kumar, and W. Zaremba, "Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research," Mar. 2018, arXiv:1802.09464 [cs]. [Online]. Available: http://doi.org/10.48550/arXiv.1802.09464

[39] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4RL: Datasets for Deep Data-Driven Reinforcement Learning," Feb. 2021, arXiv:2004.07219 [cs, stat]. [Online]. Available: http://doi.org/10.48550/arXiv.2004.07219

[40] E. Coumans and Y. Bai, "Pybullet, a Python Module for Physics Simulation for Games, Robotics and Machine Learning," 2016. [Online]. Available: https://pybullet.org/

[41] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A Physics Engine for Model-Based Control," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 5026–5033. [Online]. Available: http://doi.org/10.1109/IROS.2012.6386109

[42] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal Value Function Approximators," in Proceedings of the 32nd International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1312–1320. [Online]. Available: https://proceedings.mlr.press/v37/schaul15.html

[43] J. Ramírez, W. Yu, and A. Perrusquía, "Model-free reinforcement learning from expert demonstrations: a survey," Artificial Intelligence Review, vol. 55, no. 4, pp. 3213–3241, Apr. 2022. [Online]. Available: https://doi.org/10.1007/s10462-021-10085-1

[44] A. Shrivastava, A. Gupta, and R. Girshick, "Training Region-Based Object Detectors with Online Hard Example Mining," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 761–769. [Online]. Available: http://doi.org/10.1109/CVPR.2016.89

[45] S. C. Y. Chan, S. Fishman, A. Korattikara, J. Canny, and S. Guadarrama, "Measuring the Reliability of Reinforcement Learning Algorithms," Apr. 2020. [Online]. Available: https://iclr.cc/virtual_2020/poster_SJlpYJBKvH.html

[46] C. Colas, O. Sigaud, and P.-Y. Oudeyer, "How Many Random Seeds? Statistical Power Analysis in Deep Reinforcement Learning Experiments," Jul. 2018, arXiv:1806.08295 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1806.08295

[47] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep Reinforcement Learning That Matters," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1, Apr. 2018, number: 1. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11694

[48] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare, "Deep Reinforcement Learning at the Edge of the Statistical Precipice," in Advances in Neural Information Processing Systems, vol. 34. Curran Associates, Inc., 2021, pp. 29 304–29 320. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html

[49] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The Arcade Learning Environment: An Evaluation Platform for General Agents," Journal of Artificial Intelligence Research, vol. 47, pp. 253–279, Jun. 2013. [Online]. Available: https://jair.org/index.php/jair/article/view/10819

[50] R. S. Sutton, "Learning to Predict by the Methods of Temporal Differences," Machine Learning, vol. 3, no. 1, pp. 9–44, Aug. 1988. [Online]. Available: https://doi.org/10.1007/BF00115009

[51] D. Kumaran, D. Hassabis, and J. L. McClelland, "What Learning Systems do Intelligent Agents Need? Complementary Learning Systems Theory Updated," Trends in Cognitive Sciences, vol. 20, no. 7, pp. 512–534, Jul. 2016. [Online]. Available: https://doi.org/10.1016/j.tics.2016.05.004

[52] "Soft Actor-Critic — Spinning Up documentation." [Online]. Available: https://spinningup.openai.com/en/latest/algorithms/sac.html

[53] R. Agarwal, D. Schuurmans, and M. Norouzi, "An Optimistic Perspective on Offline Reinforcement Learning," in Proceedings of the 37th International Conference on Machine Learning. PMLR, Nov. 2020, pp. 104–114, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v119/agarwal20c.html

[54] A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, Z. D. Guo, and C. Blundell, "Agent57: Outperforming the Atari Human Benchmark," in Proceedings of the 37th International Conference on Machine Learning. PMLR, Nov. 2020, pp. 507–517, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v119/badia20a.html

DÁNIEL HORVÁTH (IEEE MEMBER) received his M.Sc. degree with highest honours in mechatronics at the Budapest University of Technology and Economics, Hungary, in 2019. As part of his master's studies, he spent one semester each at the Technical University of Denmark in Copenhagen and at the Otto von Guericke University in Magdeburg, Germany, in 2018.

He is pursuing his Ph.D. at the Eötvös Loránd University, Budapest, Hungary in computer science in collaboration with the Institute for Computer Science and Control, Budapest Hungary, and, as a Campus France scholar, with MINES Paris-PSL, in Paris, France under the supervision of Gábor Erdős, Zoltán Istenes, and Fabien Moutarde. His main research areas are reinforcement learning, curriculum learning, transfer learning, computer vision, and robotics.

JESÚS BUJALANCE MARTÍN received his M.Sc. degree in mathematics and computer vision at IP Paris - Télécom ParisTech and Paris-Saclay University - ENS Cachan (M.Sc. MVA), in 2019.

As part of his master's studies, he spent one semester at Shanghai Jiao Tong University in 2018. He is pursuing his Ph.D. at MINES Paris-PSL under the supervision of Fabien Moutarde. His main research areas are reinforcement learning, computer vision, and robotics.

GÁBOR ERDOS received his M.Sc. degree in mechanical engineering at the State University of New York at Buffalo in 1995 and his Ph.D. degree at the Budapest University of Technology and Economics, Hungary, in 2000. He was a post-doctoral researcher at the École Polytechnique Fédérale de Lausanne, Switzerland until 2002.

He joined the Institute for Computer Science and Control, Budapest, Hungary as a researcher in 2003 and since 2018, he is the deputy head of the Research Laboratory on Engineering and Management Intelligence. His main research fields are robotics, point-cloud and 3D modeling, digital twin models, multi-body kinematics, and virtual manufacturing.

ZOLTÁN ISTENES received his Ph.D. degree in Informatics in 1997 at the University of Nantes, France.

He is an associate professor at the Faculty of Informatics, Eötvös Loránd University (ELTE) in Budapest, Hungary. He established the ELTE Informatics Robotics Lab and currently directs the Erasmus Mundus Joint Master in Intelligent Field Robotic Systems (IFROS) in Hungary. At the European Institute of Innovation and Technology (EIT Digital), he manages the Budapest Doctoral Training Centre. His expertise encompasses a spectrum of fields including computer architectures, artificial intelligence, and robotics, with a recent focus on IoT, UAVs, and autonomous self-driving vehicles.

FABIEN MOUTARDE , after graduating from Ecole Polytechnique, Paris, France in 2007, received his PhD degree in astrophysics in 1991 at Paris-Diderot University, Paris VII, France, and has obtained his "Habilitation to supervise PhDs" (HdR) in Engineering Sciences in 2013 at Pierre et Marie Curie University, Paris VI, France.

He joined MinesParis-PSL in Paris, France in 1996, where he is full professor since 2015, and is currently the director of the Center for Robotics. His research is centered on artificial intelligence and machine learning for robotics, particularly computer vision and reinforcement learning, with a focus on intelligent vehicles, as well as mobile or/and collaborative robots.