

Contact Energy Based Hindsight Experience Prioritization

Erdi Sayar¹, Zhenshan Bing^{1,*}, Carlo D’Eramo^{2,3,4}, Ozgur S. Oguz⁵, Alois Knoll¹

Abstract—Multi-goal robot manipulation tasks with sparse rewards are difficult for reinforcement learning (RL) algorithms due to the inefficiency in collecting successful experiences. Recent algorithms such as Hindsight Experience Replay (HER) expedite learning by taking advantage of failed trajectories and replacing the desired goal with one of the achieved states so that any failed trajectory can be utilized as a contribution to learning. However, HER uniformly chooses failed trajectories, without taking into account which ones might be the most valuable for learning. In this paper, we address this problem and propose a novel approach Contact Energy Based Prioritization (CEBP) to select the samples from the replay buffer based on rich information due to contact, leveraging the touch sensors in the gripper of the robot and object displacement. Our prioritization scheme favors sampling of contact-rich experiences, which are arguably the ones providing the largest amount of information. We evaluate our proposed approach on various sparse reward robotic tasks and compare it with the state-of-the-art methods. We show that our method surpasses or performs on par with those methods on robot manipulation tasks. Finally, we deploy the trained policy from our method to a real Franka robot for a pick-and-place task. We observe that the robot can solve the task successfully. The videos and code are publicly available at: https://erdiphd.github.io/HER_force/.

I. INTRODUCTION

Reinforcement learning (RL) enables an agent to learn optimal behavior through trial-and-error interactions with an environment [1]–[3]. Deep RL combined with neural networks has drawn attention to sequential decision-making problems due to its success in exceeding human-level performance in playing Atari games [4], [5], beating Go master [6] and learning to achieve robotic tasks [7], [8], [9].

One of the biggest challenges in robotic tasks is to design an effective reward function [10], which requires domain knowledge and the ability to examine tasks in advance. However, defining reward functions is not trivial for certain manipulation and control tasks. Therefore, a binary reward, which indicates whether the task was accomplished successfully or not, is often used for developing RL algorithms. Unfortunately, binary rewards result in a sparse reward signal due to the insufficient amount of successful experiences. Hindsight experience replay (HER) [11] addresses the problem of sparse reward by substituting desired goals with the achieved goals sampled from failed trajectories (i.e., the sequence of interactions between the agent and the

environment). In essence, failed episodes are transformed into successful ones. Since HER samples achieved goals uniformly from the visited states, it does not take into account which visited states are most beneficial for learning, resulting in sample inefficiency. However, prioritization of trajectories may alleviate this problem. For instance, Prioritized Experience Replay (PER) [12] prioritizes trajectories based on temporal difference error, Maximum Entropy-based Prioritization (MEP) [13] leverages entropy, and Energy-Based Prioritization (EBP) [14] utilizes object energy. However, none of these methods incorporates contact feedback.

Another significant challenge in robotics is the uncertainty during object grasping. Incomplete or noisy observations of object pose or shape contribute to this uncertainty, making it harder for robots to perform stable grasps. Contact feedback is essential during manipulation and grip acquisition, whereas visual feedback is crucial for determining the grasp position. To address this challenge, several studies have explored the use of touch sensors to enhance robotic grasping. Merzic et al. [15] demonstrated the importance of contact feedback in making grasps more stable using a multi-fingered hand with contact sensing. Wang et al. [16] developed a robot system that uses tactile sensing to determine the physical characteristics of grasping objects. In the realm of RL for manipulation tasks, touch information has also been shown to improve the performance of algorithms. Melnik et al. [17] analyzed the learning performance of deep RL algorithms for dexterous in-hand object manipulation tasks with and without touch sensors and found that touch information can improve sampling efficiency and overall performance. Vulin et al. [18] used a touch sensor in the end effector of a robot and a touch-based intrinsic reward function (i.e., a contact feedback-based reward) to improve learning progress for multi-goal RL manipulation tasks. However, they introduced a contact-based reward function in addition to the binary reward and a threshold value to restrict cumulative contact forces to two empirically determined discrete values while disregarding discrepancies between contact force values.

We observe that for manipulation tasks, the robot needs to exert force and change the object’s position in order to complete the tasks successfully. Therefore, in our approach, we further extended prior research, EBP [14] and CPER [18], by incorporating object displacement with a touch sensor while exclusively relying on the binary reward signal without the need for additional reward functions. Additionally, we have addressed the uniform sampling strategy of HER by introducing a new replay buffer prioritization method based on touch sensors. Finally, we have implemented Sim2Real transfer.

¹Technical University of Munich, erdi.sayar@tum.de, {bing, knoll}@in.tum.de

²University of Würzburg, carlo.deramo@uni-wuerzburg.de

³TU Darmstadt, carlo.deramo@tu-darmstadt.de

⁴Hessian.AI, The Hessian Center for Artificial Intelligence

⁵Bilkent University, ozgur@cs.bilkent.edu.tr

Corresponding author: Zhenshan Bing

The main contributions of this paper are the following:

- Incorporating touch sensors in the robot’s end effector to explore the effectiveness of contact in robotic tasks with only relying on the binary sparse reward signal.
- Introducing a novel replay buffer prioritization scheme Contact Energy Based Prioritization (CEBP) by computing *contact energy*. This leverages touch feedback and displacement of the object. Furthermore, a sigmoid function is used for smoothing contact energy within a range, instead of restricting them to discrete values as in the CPER benchmark. We also conducted an ablation study that examines the impact of using sigmoid function with different parameters.
- We compare our approach with different benchmarks in 7-DOF Fetch Robotic-arm MuJoCo simulation environment [19]. Furthermore, we implemented Sim2Real transfer and tested our policy on a Franka robot to demonstrate the applicability of the proposed prioritization scheme.

II. RELATED WORK

A. Tactile Feedback

Tactile sensor provides local information about contact regions. Different type of tactile sensors such as capacitive [20], resistive [21] and optical-based [22] have been made in order to use for various applications such as object recognition [23], slip detection [24], pose estimation [25]. Tactile sensors have been shown to improve grasping performance and are promising for manipulation tasks [26]. Manipulation tasks are mostly challenging tasks even with vision-based methods due to occlusion and poor lighting situations. Instead, tactile sensing can provide rich information from multiple contact points [27]. These works [28], [17], [29] have proven that tactile feedback can improve the learning performance.

The works [17] and [18] inspired us to carry out this research. In the first work [17] Shadow-Dexterous-Hand are equipped with force sensors in different regions of the hand. As we explained our difference in the introduction section, in the second work [18], a touch-dependent intrinsic reward is added to the binary reward as soon as the total amount of touch value is greater than an empirically found threshold.

B. Energy-Based Hindsight Experience Prioritization

Zhao et al. [14] take into account the total trajectory energy to prioritize the experiences. The total trajectory energy consists of the potential energy, the kinetic energy, and the rotational energy of the object, and it is calculated at every timestep. The motivation for this work is that prioritizing experiences that have high trajectory energy is a more efficient approach for reinforcement learning in robotics because the robot must change the energy of an object in order to complete a task. Generally, the difficulty of a task is directly proportional to the amount of work required from the robot. Therefore, the trajectory energy of the object is used to estimate the difficulty of the task. Consequently, prioritizing experiences based on energy levels enhances the efficiency of the learning process.

C. Prioritized Experience Replay

Schaul et al. [12] examine how prioritizing certain transitions during experience replay can enhance its efficiency and effectiveness compared to replaying all transitions uniformly. The main concept of this study is that an RL agent can learn better from some transitions than others. Prioritized experience replay (PER) prioritizes experiences according to temporal difference (TD) error which is the discrepancy between the actual value and next-step bootstrap. In other words, TD-error indicates how much the RL agent needs to adjust its predictions to better approximate the true value function. An extension of this approach to multi-task RL is presented in [30]. Prioritizing transitions with higher TD errors means that the agent will focus more on learning from the experiences that it had trouble predicting accurately, which could lead to more effective learning and faster convergence to an optimal policy.

D. Maximum Entropy-based Prioritization

Maximum Entropy-based Prioritization [13] improves the sample efficiency by combining maximum entropy distribution and multi-goal RL. It encourages agents to traverse diverse goal-state trajectories while maximizing expected returns. Sampling the trajectories from a higher entropy distribution helps the agent to assign priority to those under-represented goals. The underlying idea is to prioritize the learning of those goals that are less frequently represented but have a relatively higher value in terms of what can be learned from them.

E. Hindsight Experience Replay

In environments with sparse rewards, it may take a long time to receive reward signals even after completing several sub-goals. As a result, the agent cannot gather sufficient rewards to reinforce its actions and cannot learn [31]. Andrychowicz et al. [11] developed an effective method to address the sparse reward issue by replacing the desired goals with the achieved goals from visited states which can be seen as a form of implicit curriculum learning. Because the hindsight goals in HER are selected uniformly from all visited states in the replay buffer, hindsight goals are mostly distributed around the initial state. If desired goals are far away from the initial states, HER cannot solve these tasks effectively as no reward signal is provided.

III. BACKGROUND

Multi-goal RL can be modeled as a goal-oriented Markov Decision Process (MDP) with continuous state and action spaces: $\langle \mathcal{S}, \mathcal{A}, \mathcal{G}, \mathcal{T}, \mathcal{R}, p, \gamma \rangle$ where \mathcal{S} is a continuous state space, \mathcal{A} is a continuous action space, \mathcal{G} is a set of goals, \mathcal{T} is the unknown transition function, $R(s, g)$ denotes the immediate reward obtained by an agent upon reaching state $s \in \mathcal{S}$, $p(s_0, g)$ is a joint probability distribution over initial states and desired goals and $\gamma \in [0, 1]$ is a discount factor. The reward function is defined as

$$R(s, g) = \mathbf{1}[\|\phi(s) - g\|_2 \leq \epsilon] - 1. \quad (1)$$

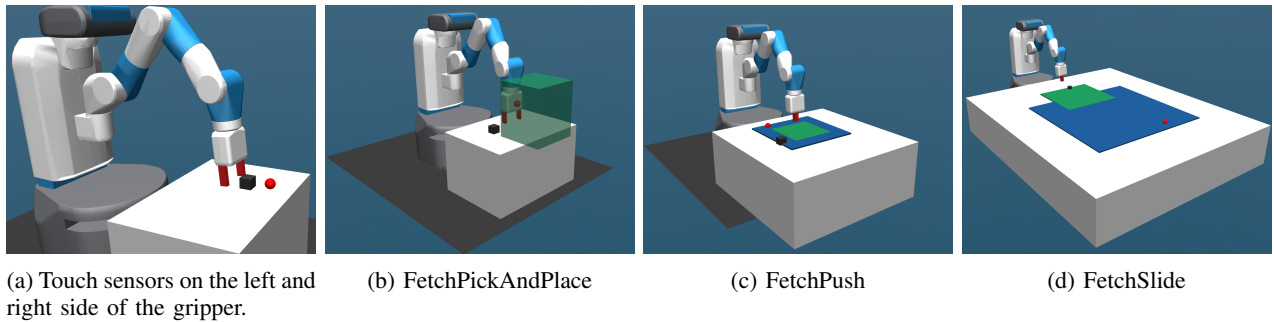


Fig. 1: Overview of the robotic manipulation benchmark tasks.

where $\mathbf{1}$ is the indicator function, ϕ is a function that maps states to goal representations and ϵ is a fixed threshold. The agent aims to determine the optimal policy π that maximizes the expected cumulative reward which can be estimated by the value function. The idea of the universal value function is to integrate goals into value function and generalize it over the goals. $V^\pi(s, g)$ is defined as a goal-based value function under a policy π for any given goal g for all state $s \in \mathcal{S}$ [32].

Within this framework, the learning problem can be represented as an RL problem that seeks a policy $\pi : \mathcal{S} \times \mathcal{G} \rightarrow \mathcal{A}$ such that the expected discounted sum of rewards is maximized for any given goal.

IV. METHODOLOGY

When we train robots to perform manipulation tasks, the sparse reward signal can hinder the agent’s learning process by making it difficult to discern which actions lead to positive outcomes, resulting in slow or inefficient policy learning. HER leverages failed experiences by replacing the desired goal with the achieved goal of the failed experiences. With this modification, any failed experience returns a nonnegative reward. However, if the initial position of the robot is far from the desired position, hindsight goals generated by HER may not be effective in solving the task because they are sampled randomly from the visited states, which are mostly distributed around the initial state. Consequently, experiences are not regarded based on their importance for learning, which leads to sample inefficiency.

As we are interested in robot manipulation tasks, we leverage the touch sensor and prioritize experiences based on the contact forces occurring between the gripper and the object during the interaction, as well as the object displacement. The contact forces can be used to prioritize experiences that involve more interaction with the gripper, indicating that the robot is indeed manipulating the object. Nevertheless, relying solely on contact force as a criterion for prioritization might be insufficient or even misleading as the gripper may engage with the object in impractical ways, such as exerting downward pressure onto the object’s surface. In such cases, the touch sensor detects substantial forces, yet

the experiences resulting from the robot’s actions fail to yield useful outcomes in manipulating objects. Moreover, these experiences could inadvertently lead such a prioritization algorithm to overemphasize them in the learning process, as larger contact forces would be sampled with a higher probability of learning.

Additionally, experiences involving collisions between the robot and other objects in its vicinity, like a table, tend to yield unfavorable results. Consequently, introducing object displacement as a complementary consideration to contact forces serves to eliminate the prioritization of infeasible scenarios. In addition, to diminish the adversarial effect of collisions due to high forces, we employ a sigmoid function to smooth the contact energy, i.e., the extreme forces are saturated. Furthermore, unlike CPER [18], where cumulative, continuous contact forces are converted only to two discrete values $\{1, \lambda\}$, thereby dismissing the significance of contact forces except for these predefined values in the set, the sigmoid function allows us to prioritize experiences based on a continuous range. In the ablation study, we investigated the impact of the sigmoid function with different parameters on the learning process.

A. Contact Energy Prioritization

The touch sensors, as shown in Fig. 1a, are used on the gripper’s left and right sides to measure contact forces. The output of the touch sensor is a non-negative scalar value, which is calculated by summing up the normal forces from all the contact points. Then, contact forces from the left and right sides are summed up into a single value. Next, in each time step, we multiply these contact forces by the Euclidian norm of the object’s displacement and calculate the contact energy as

$$c(e, t) = \sum_{i=0}^t (f_{\text{left}}(e, i) + f_{\text{right}}(e, i)) \cdot \|\Delta d_c(i)\|_2. \quad (2)$$

where e is episode number and $\|\Delta d_c(i)\|_2$ is the Euclidian distance of the object’s displacement between current i and previous timestep $i - 1$.

The touch sensor relies on contact forces, which can be large during exploration when the gripper collides with objects around it, such as a table. The higher the robot’s velocity when it collides with objects, the higher the contact

¹Benchmark tasks are used exactly as in https://github.com/nikolavulin/tactile_intrinsic_motivation/tree/master/Gym [18].

²All methods are trained using the same sparse reward setting.

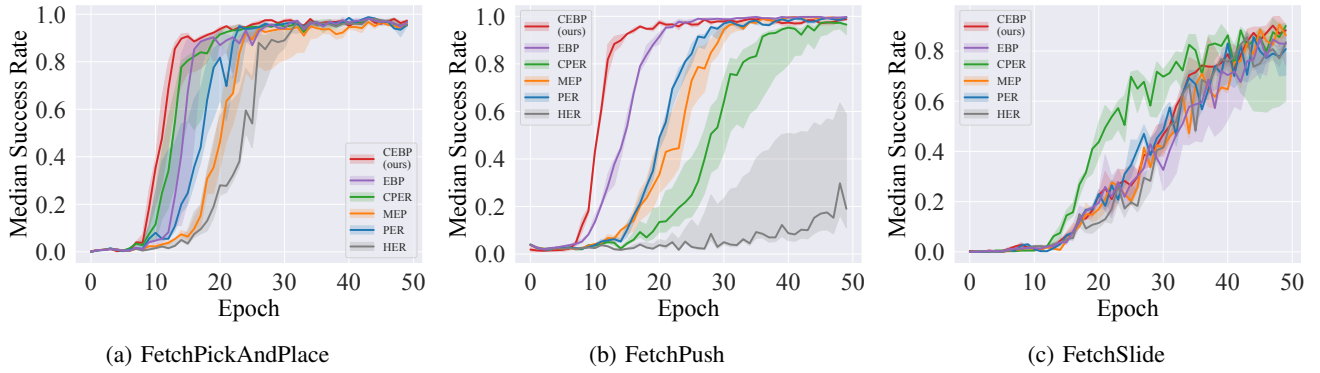


Fig. 2: Median success rate for all three Fetch tasks. The average success rate (line) and interquartile range (shaded) are shown with training 5 random seeds. ²

force values will be. However, these experiences with large contact forces are mostly failed trajectories and lack valuable information. As a result, they mislead our replay buffer prioritization algorithm into prioritizing them for learning, as experiences with high contact forces are sampled with higher probability. To address this issue, we are using a sigmoid function to smooth large contact values due to collision.

$$\sigma(x) = \frac{k}{1 + e^{-xT}}. \quad (3)$$

where k is the scale and T is the temperature.

$$\tilde{c}(e, t) = \sigma(c(e, t)). \quad (4)$$

where \tilde{c} is the smoothed contact energy.

Then the probability distribution of the episodes in equation (5) is calculated based on their contact-energy rich information, and a mini-batch \mathcal{B} is sampled from the replay buffer \mathcal{R} according to the probability distribution $p_{\text{episode}}(e)$

$$p_{\text{episode}}(e) = \frac{\sum_{t=0}^{\mathcal{T}} \tilde{c}(e, t)}{\sum_{i_e=0}^e \sum_{t=0}^{\mathcal{T}} \tilde{c}(i_e, t)}. \quad (5)$$

Prioritizing the experiences according to the given methods above introduces bias because it changes the distribution, thereby changing the ultimate solution that the estimates will converge to. This bias can be removed by using the importance-sampling (IS) weights given in equation (6)

$$w_i = (N \cdot p_{\text{episode}}(e))^{-\beta}. \quad (6)$$

Then, these weights are multiplied with TD error δ and the Q-value is updated. For stability reasons, weights are normalized by $1/\max(w)$ [12].

Algorithm 1 outlines our method. It can be used in combination with any off-policy algorithm, and we choose the Deep Deterministic Policy Gradients (DDPG) [33] algorithm for comparing the results in section V. Then, a desired goal is sampled, and trajectories are collected by running the policy. If the end state of the episode is chosen as a hindsight goal, then the desired goal g should be replaced with this achieved goal, and the reward function should be recalculated. The probability distribution $p_{\text{episode}}(e)$ is calculated

using equation (5) and a mini-batch is sampled according to $p_{\text{episode}}(e)$ from the replay buffer \mathcal{R} . The importance sampling weight (eq. 6) and the TD-error are then calculated for each transition sampled from the mini batch. Finally, the weights are updated.

Algorithm 1 Contact Energy Based Prioritization (CEBP)

```

An off-policy algorithm  $\mathbb{A}$  ▷ In our case  $\mathbb{A}$  is DDPG
 $\mathcal{R} \leftarrow \emptyset$ 
for  $episode = 1, M$  do
  Sample a desired goal  $g$  and an initial state  $s_0$ 
  for  $t = 0, \mathcal{T} - 1$  do ▷ Rollout one episode
    Collect a trajectory by running the policy from  $\mathbb{A}$  and
    save to the replay buffer  $\mathcal{R}$ ;
     $a_t = \pi(s_t, g)$ ;
     $r_t = r(s_t, a_t, g)$ ;
     $\mathcal{R}_i \leftarrow (s_t, a_t, r_t, s_{t+1}, g)$  ▷ Store transition
     $G := \mathcal{S}(\text{current episode})$  ▷ Sample a set of additional
    goals for replay
    for  $g' \in G$  do ▷ See HER [11]
       $r'_t = r(s_t, a_t, g')$  ▷ Recompute reward
       $\mathcal{R}_i \leftarrow (s_t, a_t, r'_t, s_{t+1}, g')$  ▷ Store transition
    end for
  end for
  Compute  $p_{\text{episode}}(e)$  using Eq.(5)
  Sample a minibatch  $\mathcal{B} : \{(s_t, a_t, s_{t+1}, r_t, g)_i\}_i^N \sim \mathcal{R}$ ;
  for  $j = 1, k$  do
     $(s_j, a_j, s_{j+1}, r_j, g) \leftarrow \mathcal{B}(j)$ 
    Compute IS weight:
     $w_j = (N \cdot p_{\text{episode}}(e))^{-\beta} / \max(w)$ 
    Compute TD Error:
     $\delta_j = r_j + \gamma_j Q_{\text{target}}(s_j, g, \arg\max_a(Q' - Q))$ 
    Accumulate weight-change:
     $\Delta \leftarrow \Delta + w_j \delta_j \nabla_{\theta} Q$ 
  end for
  Update weights
   $\theta \leftarrow \tau \cdot \theta + (1 - \tau) \cdot \theta'$ 
end for

```

V. EXPERIMENT

This section describes the robot simulation benchmark used for evaluating the proposed approach. We test the proposed replay buffer using a 7-DOF robotic arm [34] in the MuJoCo simulation [35]. The robotic arm environment is designed as a standard benchmark for Multi-Goal RL [19].

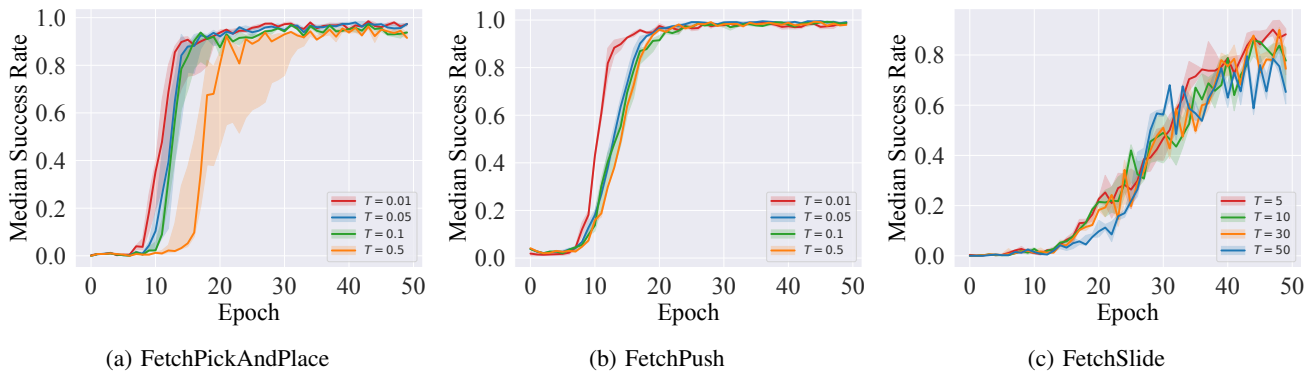


Fig. 3: Through the ablation study, we analyze the impact of different temperature parameters T of sigmoid function on the success rate. The median success rate (line) and interquartile range (shaded) are shown with training 5 random seeds.

Three standard manipulation tasks from the benchmark, pick-and-place, push, and slide, are shown in Fig.1 and cloned from CPER where the original goal space of HER (green) is extended to a larger area (blue) in order to increase the difficulty of the tasks. Furthermore, the goals in the *FetchPickAndPlace* are uniformly sampled to further increase the task difficulty (instead of using 50% of targets on the table as in the HER benchmark task.)

The state vector consists of the goal position as well as the position, orientation, linear and angular velocity of the end-effector and the object. A goal represents the desired position for an object. It is assumed that the task is accomplished if the object reaches the goal within the distance set by a threshold (i.e., defined as in equation (1)). If the object is not in the range of the goal, the agent receives a negative reward -1 ; otherwise, the reward is 0. We compare the proposed approach CEBP, with 5 different random seeds, against other methods, namely CPER [18], PER [12], MEP [13], EBP [14] and HER [11] using three robotic manipulation tasks shown in Fig. 1.

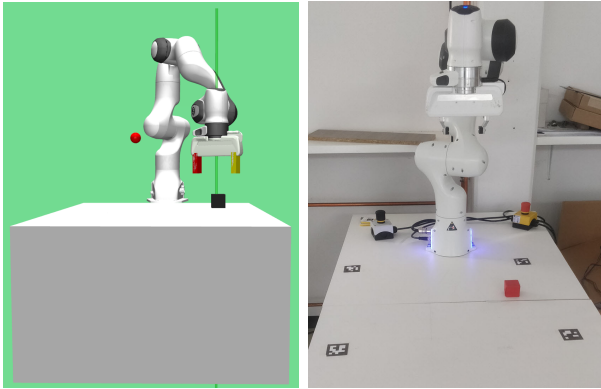
- Pick & Place Task (*FetchPickAndPlace-v1*): The goal is to grasp the black object with a gripper and place it at the target location shown as a red dot in Fig. 1b. The target positions are sampled from the green shaded region.
- Push Task (*FetchPush-v1*): The position of both the target and the black object are sampled from the blue area on the table, and the robot’s gripper is clamped as illustrated in Fig. 1c. The goal is to push the object to the target position.
- Slide Task (*FetchSlide-v1*): The black object is sampled and placed on a long slippery table and the target is sampled from the blue area and it is outside of the robot’s workspace (Fig. 1d). The goal is to hit the puck with a specific force so that it slides on the table and reaches the target location.

The outcomes of the simulation shown in Fig. 2 provide insight into the efficacy of the contact energy-based prioritization (CEBP) that has been proposed. The agent, trained with the CEBP method, consistently outperforms existing benchmarks in both pick-and-place and push tasks. The agent converges to a near-optimal policy after 20 epochs

and 16 epochs for the pick-and-place (Fig.2a) and push tasks (Fig.2b), respectively. Especially, in the push task, our approach quickly finds out optimum actions at the early stages of training. In the slide task (Fig.2c), CPER initially starts to learn faster, but as the training process approaches the final phases, our approach achieves a similar final performance with a significantly lower standard deviation than CPER. On the other hand, EBP considers only the object’s energy and exhibits suboptimal performance compared to CEBP, as the tasks require object manipulation by the robot. The experimental results demonstrate that considering both object displacement and contact forces together for prioritizing the experiences results in better performance. Furthermore, prioritizing experiences based on TD-error (PER) and a distribution with higher entropy (MEP) results in suboptimal outcomes. Notably, uniform prioritization of experiences (HER) exhibits the poorest performance among the considered methods.

When we analyzed the contact forces during the pick-and-place task, we observed that their magnitudes remained relatively constant when the object was being moved within the gripper towards the target goal. However, variations in the contact forces are observed during the picking up and the placing of the object. Consequently, the difference between using continuous contact forces (as in our method) and only two discrete values (as in CPER) for prioritizing experience appears to yield marginal differences, as shown in Fig. 2a. However, in the push task, the robot pushes the object, which generates a wider range of contact forces than the other tasks. Therefore, considering contact forces with a continuous range for prioritizing experience can make a significant difference in the learning rate compared to CPER, as shown in Fig. 2b. In the slide task, contact between the robot and the object occurs for a very short time, resulting in limited contact feedback, and our method performs on par with the benchmarks.

The suboptimal performance of CPER [18] can be attributed to the prioritization strategy. During the prioritization in the CPER, the force values are accumulated through the timesteps, and when this cumulative sum surpasses a predetermined threshold, the corresponding forces are given



(a) Franka Robot simulation (b) Franka Robot real setup

Fig. 4: Franka robot setups

a constant value of λ ; otherwise, they are set to 1. As a result, a wide range of force values is compressed into a set with only two elements, $\{1, \lambda\}$. Consequently, CPER does not consider varying levels of importance of the different force values for prioritization of the experiences. In contrast to the CPER, we prioritize the experiences based on their wide range of contact-rich information using the sigmoid function.

In the ablation study, we investigate and compare the impact of temperature parameter T of the sigmoid function $\sigma(x) = \frac{k}{1+e^{-xT}}$ on the performance where k is the scale and T is the temperature parameter. We empirically found that parameters $k = 100$ and $T = 0.01$ for pick-and-place and push tasks and $k = 100$ and $T = 5$ for slide task lead to the best performance. The aforementioned issue in CPER can be confirmed and inferred from the training outcomes in the ablation study, illustrated in Fig. 3. If we increase the temperature parameter T of the sigmoid function, the curve becomes much flatter, less steeper and less sensitive to contact changes and starts to behave like a binary switch and becomes gradually similar to the approach as in CPER. Accordingly, the success rate is decreasing based on the sensitivity in the sigmoid curve.

VI. SIM2REAL

We trained the Franka robot in the MuJoCo simulation environment using a contact energy-based prioritization, shown in Fig. 4. The experiment was set up with a camera and ArUco markers. ArUco markers and the red cube object are detected using the Aruco module and a red filter with the Open-CV library, respectively, and the centers of their pixel positions are obtained as shown in Fig. 5. The policy that has been trained is directly transferred from simulation to the real-world setting and outputs the linear motion in Cartesian space relative to its end-effector position as well as the state of the gripper gap. The parameters of linear motion are given to the Franka robot. Hereby, we robustly transferred the policy trained from the simulation environment into the real world.

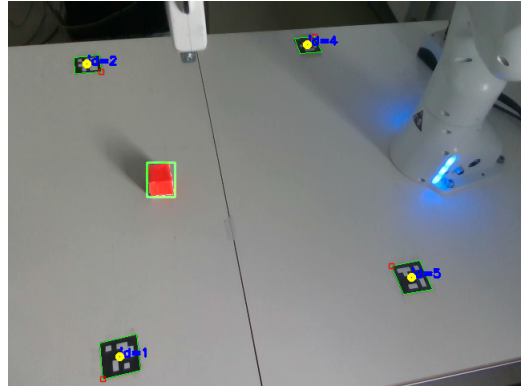


Fig. 5: The camera view, showing the identified ArUco markers and detected object.

VII. CONCLUSIONS

Limitations we faced during the Sim2Real transfer are the occlusion for the push task and a friction parameter discrepancy between simulation and real world settings in the slide task. Therefore, we were not able to execute the policy robustly in the real world setting in these tasks. However, we believe that the occlusion and friction problems can be overcome using multiple cameras from different perspectives and estimating the real world friction parameter, and using it in the simulation environment, respectively.

Our study introduces a new prioritization scheme named *Contact Energy Based Prioritization (CEBP)*, which is designed for multi-goal RL with sparse rewards. We evaluated the efficacy of this approach in the MuJoCo simulation environment, testing it across three different robotic manipulation tasks. Experimental results indicate that this contact energy-based prioritization not only enhances performance but also improves exploration across three fundamental manipulation tasks. Furthermore, the ablation study investigates the effect of the temperature parameter of the sigmoid function on the success rate. The suboptimal performance of the benchmark CPER is confirmed by the training results in the ablation study. Additionally, we selected the pick-and-place task for Sim2Real transfer using a Franka robot. Based on our analysis, we can infer that the utilization of tactile feedback in conjunction with object displacement could serve as a catalyst for enhancing reinforcement learning performance, especially for contact-rich robot manipulation tasks.

For future work, we aim to investigate applying the concept of sparse reward and hindsight experience into meta-reinforcement learning (meta-RL) agents [36]–[41].

VIII. ACKNOWLEDGEMENTS

This research has been supported by the European Union’s Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No.945539 (Human Brain Project SGA3) and German Federal Ministry of Education and Research (BMBF) (Project: 01IS22078). This work was also funded by Hessian.ai through the project ‘The Third Wave of Artificial Intelligence – 3AI’ by the Ministry for Science and Arts of the state of Hessen.

REFERENCES

- [1] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, 1996.
- [2] Z. Bing, C. Lemke, L. Cheng, K. Huang, and A. Knoll, "Energy-efficient and damage-recovery slithering gait design for a snake-like robot based on reinforcement learning and inverse reinforcement learning," *Neural Networks*, vol. 129, pp. 323–333, 2020.
- [3] Z. Bing, L. Cheng, K. Huang, and A. Knoll, "Simulation to real: Learning energy-efficient slithering gaits for a snake-like robot," *IEEE Robotics & Automation Magazine*, vol. 29, no. 4, pp. 92–103, 2022.
- [4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.
- [5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., "Human-level control through deep reinforcement learning," *nature*, 2015.
- [6] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al., "Mastering the game of go with deep neural networks and tree search," *nature*, 2016.
- [7] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, "Autonomous inverted helicopter flight via reinforcement learning," in *Experimental robotics IX: The 9th international symposium on experimental robotics*. Springer, 2006, pp. 363–372.
- [8] Z. Bing, H. Zhou, R. Li, X. Su, F. O. Morin, K. Huang, and A. Knoll, "Solving robotic manipulation with sparse reward reinforcement learning via graph-based diversity and proximity," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 3, pp. 2759–2769, 2023.
- [9] Z. Bing, M. Brucker, F. O. Morin, R. Li, X. Su, K. Huang, and A. Knoll, "Complex robotic manipulation via graph-based hindsight goal generation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 12, pp. 7863–7876, 2022.
- [10] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *ICML*, 1999.
- [11] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," *CoRR*, vol. abs/1707.01495, 2017. [Online]. Available: <http://arxiv.org/abs/1707.01495>
- [12] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.
- [13] R. Zhao, X. Sun, and V. Tresp, "Maximum entropy-regularized multi-goal reinforcement learning," in *International Conference on Machine Learning*, 2019, pp. 7553–7562.
- [14] R. Zhao and V. Tresp, "Energy-based hindsight experience prioritization," in *Conference on Robot Learning*. PMLR, 2018, pp. 113–122.
- [15] H. Merzic, M. Bogdanovic, D. Kappler, L. Righetti, and J. Bohg, "Leveraging contact forces for learning to grasp," 2018.
- [16] C. Wang, S. Wang, B. Romero, F. Veiga, and E. Adelson, "Swing-bot: Learning physical features from in-hand tactile exploration for dynamic swing-up manipulation," 2021.
- [17] A. Melnik, L. Lach, M. Plappert, T. Korthals, R. Haschke, and H. Ritter, "Tactile sensing and deep reinforcement learning for in-hand manipulation tasks," in *IROS Workshop on Autonomous Object Manipulation*, 2019.
- [18] N. Vulin, S. Christen, S. Stevsic, and O. Hilliges, "Improved learning of robot manipulation tasks via tactile intrinsic motivation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, p. 2194–2201, Apr. 2021. [Online]. Available: <http://dx.doi.org/10.1109/LRA.2021.3061308>
- [19] 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," 2018.
- [20] T. Li, H. Luo, L. Qin, X. Wang, Z. Xiong, H. Ding, Y. Gu, Z. Liu, and T. Zhang, "Flexible capacitive tactile sensor based on micropatterned dielectric layer," *Small*, vol. 12, no. 36, pp. 5042–5048, 2016.
- [21] K. Weiß and H. Worn, "The working principle of resistive tactile sensor cells," in *IEEE International Conference Mechatronics and Automation*, 2005, vol. 1. IEEE, 2005, pp. 471–476.
- [22] E. Donlon, S. Dong, M. Liu, J. Li, E. Adelson, and A. Rodriguez, "Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1927–1934.
- [23] M. Kaboli, K. Yao, D. Feng, and G. Cheng, "Tactile-based active object discrimination and target object search in an unknown workspace," *Autonomous Robots*, vol. 43, no. 1, pp. 123–152, 2019.
- [24] J. W. James, N. Pestell, and N. F. Lepora, "Slip detection with a biomimetic tactile sensor," *Robotics and Automation Letters*, 2018.
- [25] J. Bimbo, S. Luo, K. Althoefer, and H. Liu, "In-hand object pose estimation using covariance-based tactile to geometry matching," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 570–577, 2016.
- [26] J. M. Romano, K. Hsiao, G. Niemeyer, S. Chitta, and K. J. Kuchenbecker, "Human-inspired robotic grasp control with tactile sensing," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1067–1079, 2011.
- [27] M. Kaboli, "New methods for active tactile object perception and learning with artificial robotic skin," Dissertation, Technische Universität München, München, 2017.
- [28] H. Van Hoof, T. Hermans, G. Neumann, and J. Peters, "Learning robot in-hand manipulation with tactile features," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2015.
- [29] A. Koenig, Z. Liu, L. Janson, and R. Howe, "Tactile grasp refinement using deep reinforcement learning and analytic grasp stability metrics," 2021.
- [30] C. D'Eramo and G. Chalvatzaki, "Prioritized sampling with intrinsic motivation in multi-task reinforcement learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.
- [31] A. Zai and B. Brown, *Deep reinforcement learning in action*. Manning Publications, 2020.
- [32] T. Schaul, D. Horgan, K. Gregor, and D. Silver, "Universal value function approximators," in *ICML*, 2015.
- [33] 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," *arXiv preprint arXiv:1509.02971*, 2015.
- [34] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," 2016.
- [35] 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*. IEEE, 2012, pp. 5026–5033.
- [36] Z. Bing, L. Knak, L. Cheng, F. O. Morin, K. Huang, and A. Knoll, "Meta-reinforcement learning in nonstationary and nonparametric environments," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [37] Z. Bing, A. Koch, X. Yao, K. Huang, and A. Knoll, "Meta-reinforcement learning via language instructions," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5985–5991.
- [38] M. Wang, Z. Bing, X. Yao, S. Wang, H. Kai, H. Su, C. Yang, and A. Knoll, "Meta-reinforcement learning based on self-supervised task representation learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 10 157–10 165.
- [39] X. Yao, Z. Bing, G. Zhuang, K. Chen, H. Zhou, K. Huang, and A. Knoll, "Learning from symmetry: Meta-reinforcement learning with symmetrical behaviors and language instructions," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 5574–5581.
- [40] Z. Bing, D. Lerch, K. Huang, and A. Knoll, "Meta-reinforcement learning in non-stationary and dynamic environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3476–3491, 2022.
- [41] H. Zhou, X. Yao, Y. Meng, S. Sun, Z. Bing, K. Huang, and A. Knoll, "Language-conditioned learning for robotic manipulation: A survey," *arXiv preprint arXiv:2312.10807*, 2023.