Evaluating DDPG, TD3, and SAC in Panda-Gym: The Impact of Hindsight Experience Replay and Single Q-Networks

Zhang Ye[*]
ye.zhang1@northeastern.edu
Northeastern University

Ren Yin[*]
yin.r1@northeastern.edu
Northeastern University

April 17, 2025

Abstract

In this report, we investigate the performance of three off-policy deep reinforcement learning algorithms—DDPG [11], TD3 [5], and SAC [8]—on goal-conditioned robotic manipulation tasks using the Panda-Gym simulation suite [6]. Our study centers on two critical design elements: the use of Hindsight Experience Replay (HER) [1], and the role of the Double Q-learning trick [9, 16] in value estimation.Through extensive experiments, we confirm that HER substantially improves learning efficiency in sparse-reward environments across all algorithms. Furthermore, we observe that removing the Double Q mechanism—i.e., relying on a single Q-network—can markedly accelerate convergence and enhance final success rates, particularly in conjunction with HER. These results reveal a counterintuitive interaction: in HER-augmented settings, the conservative updates induced by Double Q-learning may actually impede performance by introducing unnecessary value underestimation. Our findings call for a reevaluation of the Double Q design choice in HER-based goal-conditioned learning and suggest new directions for developing more adaptive value estimation strategies in reinforcement learning. Our code is available at https://github.com/evenjohnyz/CS5180.

Keywords

Reinforcement Learning; Robotic Manipulation; Hindsight Experience Replay (HER); DDPG; TD3; SAC

1 Introduction

Reinforcement Learning (RL) has shown great promise in enabling robotic agents to autonomously acquire complex manipulation skills. However, many real-world and simulated robotics tasks, such as reaching or pick-and-place, suffer from sparse reward signals, where the agent receives meaningful feedback only upon task completion.[12, 7, 13] This sparsity often hinders effective learning, particularly in early stages when successful episodes are rare.

To address this, Hindsight Experience Replay (HER) [1] was proposed as a powerful augmentation strategy. By relabeling failed attempts with achieved goals, HER transforms otherwise uninformative trajectories into valuable learning experiences. This technique has been widely adopted in goal-conditioned RL settings and has demonstrated consistent improvements across various environments.

Despite these advances, state-of-the-art algorithms in deep reinforcement learning such as TD3[5] and SAC[8] incorporate the Double Q trick[16] to mitigate

[*] Both authors contributed equally to this work.

value overestimation. While effective in many continuous control tasks, we observed that Double Q can, in some settings, lead to overly conservative updates and slow down learning—especially when combined with HER in sparse reward domains.[4]

To explore the effectiveness of deep reinforcement learning in complex environments, and to understand the impact of Hindsight Experience Replay (HER) and the Double Q-learning trick, we systematically evaluate three off-policy actor-critic algorithms—DDPG, TD3, and SAC—on the **PandaReach** and **PandaPickAndPlace** environments from the Panda-Gym suite. In general, **PandaReach** is considered a relatively easier environment compared to **PandaPickAndPlace**, as the latter requires more precise control and coordination to successfully complete the task.

In summary, our work makes the following contributions:

1. We empirically demonstrate that Hindsight Experience Replay (HER) significantly improves learning performance across all three algorithms, especially in sparse-reward environments.
2. We show that removing the Double Q trick—i.e., using a single critic network—can accelerate learning and improve final success rates. This effect is particularly pronounced in SAC and TD3, across both simple and complex tasks.
3. We identify and highlight a previously overlooked trade-off between conservative value estimation and learning efficiency in goal-conditioned reinforcement learning.

2 Background

2.1 Markov Decision Process

Reinforcement Learning (RL) provides a formal framework for sequential decision-making under uncertainty. It is typically modeled as a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where:

- $\mathcal{S}$: the state space
- $\mathcal{A}$: the action space
- $P(s' \mid s, a)$: the transition probability of moving from state s to state s' given action a
- $R(s, a)$: the reward function
- $\gamma \in [0, 1)$: the discount factor

At each timestep t, the agent interacts with the environment by selecting an action according to a policy π, with the objective of maximizing the expected discounted return:

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r_t \right]$$

2.2 Policy Gradient Method

After formulating the environment, the agent learns through exploration by improving a policy, which represents the strategy it follows. In policy gradient methods, the policy is updated directly by optimizing a stochastic policy $\pi_\theta(a \mid s)$, parameterized by θ. The objective is to maximize the expected return $J(\pi_\theta)$, and the update is performed using the policy gradient theorem[15]:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s \sim d^\pi,\ a \sim \pi_\theta} \left[\nabla_\theta \log \pi_\theta(a \mid s)\, Q^\pi(s, a) \right]$$

Here, $d^\pi(s)$ denotes the discounted state visitation distribution under policy π, and $Q^\pi(s, a)$ is the action-value function.

To reduce the variance of gradient estimates, actor-critic methods approximate the action-value function $Q^\pi(s, a)$ using a learned critic network. The actor (i.e., the policy) is then updated using this learned approximation, enabling more stable and sample-efficient learning.

2.3 Deep Deterministic Policy Gradient (DDPG)

Deep Deterministic Policy Gradient (DDPG) [11] is an off-policy, deterministic actor-critic algorithm designed for continuous action spaces. It maintains the following components:

- A deterministic policy (actor): $\mu_\theta(s)$
- A Q-function approximator (critic): $Q_\phi(s, a)$
- Target networks for both actor and critic, which are updated via soft updates:

$$\theta_{\text{target}} \leftarrow \tau\theta + (1 - \tau)\theta_{\text{target}}$$
$$\phi_{\text{target}} \leftarrow \tau\phi + (1 - \tau)\phi_{\text{target}}$$

The critic is trained using the Bellman target:

$$y = r + \gamma Q_{\phi'}(s', \mu_{\theta'}(s'))$$

The actor is updated by applying the deterministic policy gradient:

$$\nabla_\theta J \approx \mathbb{E}_s \left[\nabla_a Q_\phi(s, a)\big|_{a=\mu_\theta(s)} \nabla_\theta \mu_\theta(s) \right]$$

DDPG leverages experience replay and target networks to stabilize training and improve sample efficiency in high-dimensional continuous control tasks.

2.4 Twin Delayed Deep Deterministic Policy Gradient (TD3)

Twin Delayed Deep Deterministic Policy Gradient (TD3) [4] improves upon DDPG by addressing overestimation bias in Q-learning through three key modifications:

1. **Twin Q-networks:** Two independent critics Q_1 and Q_2 are trained, and the minimum of the two is used for target value estimation to mitigate overoptimistic value estimates.
2. **Delayed policy updates:** The actor and target networks are updated less frequently (e.g., every two critic updates) to improve stability.
3. **Target policy smoothing:** Noise is added to target actions to regularize value estimates and prevent the critic from exploiting sharp Q-function peaks:

$$a' = \mu_{\theta'}(s') + \epsilon$$
$$\epsilon \sim \text{clip}(\mathcal{N}(0, \sigma), -c, c)$$

The Bellman target is computed using the minimum of the two Q-values:

$$y = r + \gamma \min_{i=1,2} Q_i(s', a')$$

These techniques collectively enable more conservative and stable value estimation, significantly improving learning performance in continuous control environments.

2.5 Soft Actor-Critic (SAC)

Soft Actor-Critic (SAC) [8] introduces entropy regularization into the reinforcement learning objective to encourage exploration and maintain policy stochasticity. The objective function is defined as:

$$J(\pi) = \sum_t \mathbb{E}_{(s_t, a_t) \sim \pi} \left[r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot \mid s_t)) \right]$$

Here, $\mathcal{H}(\pi(\cdot \mid s_t))$ denotes the entropy of the policy at state s_t, and α is a temperature parameter that controls the trade-off between reward maximization and entropy.

SAC maintains the following components:

- Two soft Q-functions: $Q_1(s, a)$ and $Q_2(s, a)$
- A stochastic policy (actor): $\pi_\theta(a \mid s)$
- A value function: $V_\psi(s)$, trained to match the entropy-regularized Q-value

The value function is optimized using the following target:

$$V_\psi(s) \approx \mathbb{E}_{a \sim \pi_\theta} \left[Q(s, a) - \alpha \log \pi_\theta(a \mid s) \right]$$

The soft Bellman backup for training the Q-functions is given by:

$$y = r + \gamma \left(\min_{i=1,2} Q_i(s', a') - \alpha \log \pi(a' \mid s') \right)$$

By integrating entropy into the objective and value estimates, SAC achieves more stable and exploratory learning, making it particularly effective in continuous control tasks.

2.6 Hindsight Experience Replay (HER)

Hindsight Experience Replay (HER) [1] enhances off-policy reinforcement learning by introducing relabeled experiences to improve sample efficiency in sparse-reward settings.

Given a transition (s, a, s', g), where g is the original goal, HER selects a future state s_{future} from the same episode and defines a new goal $g' = f(s_{\text{future}})$. This synthetic goal g' replaces g in the replay buffer, effectively transforming unsuccessful attempts into successful experiences for an alternative goal.

The relabeled reward function is defined as:

$$R(s, a, g') = \begin{cases} 0, & \text{if } \|s' - g'\| < \varepsilon \\ -1, & \text{otherwise} \end{cases}$$

This relabeling strategy provides more frequent learning signals, significantly accelerating training in environments where rewards are otherwise sparse and difficult to obtain.

2.7 Double Q and Single Q

Double Q-learning [9, 16] was originally proposed to mitigate overestimation bias in Q-learning. In algorithms such as TD3 and SAC, this is implemented by using two critic networks and minimizing over their outputs:

$$y = r + \gamma \cdot \min\left(Q_1(s', a'),\ Q_2(s', a')\right)$$

While this conservative value estimation is effective in reducing overoptimism, recent findings [**?**] suggest that, when combined with Hindsight Experience Replay (HER), it can negatively impact learning by underestimating the value of rare successful transitions—thus weakening the learning signal in sparse-reward environments.

In this work, we investigate the performance implications of removing the minimum operator and instead using a **Single Q** formulation:

$$y = r + \gamma Q_1(s', a')$$

This simplification reduces underestimation and, in some settings, accelerates convergence and improves final task success rates, particularly when HER is applied.

2.8 Related Work

Among the many off-policy actor-critic algorithms proposed for continuous control, Deep Deterministic Policy Gradient (DDPG), Twin Delayed DDPG (TD3), and Soft Actor-Critic (SAC) have become foundational due to their sample efficiency and scalability. DDPG, while effective in simpler environments, often suffers from unstable training in sparse-reward settings due to its reliance on deterministic policies and sensitivity to exploration noise. TD3 improves upon DDPG by introducing three key modifications: twin Q-networks to mitigate overestimation, delayed policy updates for stability, and target policy smoothing to regularize the critic. SAC further advances this direction by adopting a stochastic policy and incorporating an entropy regularization term, which promotes more exploratory behavior and robust learning. Both TD3 and SAC benefit from the Double Q-learning trick, which encourages conservative value estimation and has been shown to improve performance across many benchmarks.

However, recent studies [14] suggest that in sparse-reward environments—particularly when Hindsight Experience Replay (HER) is applied—the conservative nature of Double Q-learning may lead to underestimation bias. This can weaken the learning signal, especially for rare but important successful transitions relabeled by HER. Despite these observations, the interaction between HER and Double Q remains underexplored. In this work, we provide empirical evidence that removing the Double Q mechanism—i.e., adopting a Single Q formulation—can lead to faster convergence and improved performance in HER-augmented goal-conditioned reinforcement learning tasks.

3 Experiment

To evaluate the impact of Hindsight Experience Replay (HER) and the removal of the Double Q-learning mechanism under different settings and tasks, we conducted experiments using two environments from the Panda-Gym suite: **PandaReach-v3** and **PandaPickAndPlace-v3**.

3.1 Environments

We evaluate the algorithms in two goal-conditioned robotic manipulation environments from the Panda-Gym suite: **PandaReach-v3** and **PandaPickAndPlace-v3**.

- **PandaReach-v3** is a relatively simple task in which the 7-DoF Franka Emika Panda robot arm must move its end-effector to a randomly sampled target position in 3D space. The reward is sparse: the agent receives a reward of 0 when the goal is reached within a 5 cm threshold, and -1 otherwise.

- **PandaPickAndPlace-v3** is a more complex task that requires the robot to first grasp a cube and then place it at a target location. This environment involves contact dynamics, grasp planning, and more intricate state transitions.

Both environments adopt the multi-goal reinforcement learning (RL) framework and return observations in the form of a dictionary with the following keys:

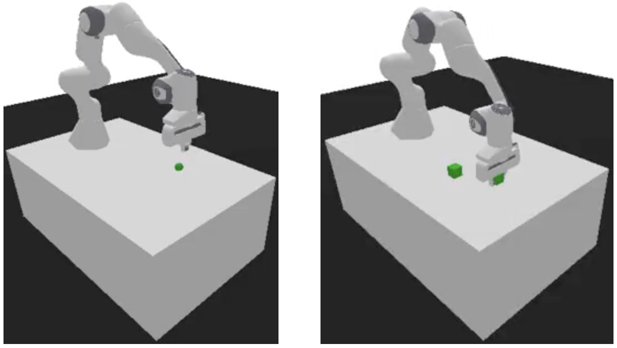
- `observation`: the current robot state

- `achieved_goal`: the goal currently achieved by the robot

- `desired_goal`: the target goal the agent is tasked with achieving

The action space is continuous and four-dimensional: $[dx, dy, dz, \texttt{grip\_action}]$, representing position deltas and the gripper command.

Figure 1 illustrates the visual difference between the two environments. In PandaReach (Figure 1a), the scene contains a single green marker indicating the target position for the robot arm to reach. In contrast, PandaPickAndPlace (Figure 1b) includes two green markers: one for the goal position and another for the object (cube) that must be grasped. The latter is clearly more challenging, requiring precise motion planning, grasping, and more nuanced interaction with the environment.



(a) PandaReach (b) PandaPickAndPlace

Figure 1: Visual comparison of Panda-Gym environments used in this study

3.2 Algorithms and Variants

We primarily evaluate three off-policy actor-critic algorithms: DDPG, TD3, and SAC. Each algorithm is tested in two variants to assess the effect of the Double Q-learning mechanism in the presence of Hindsight Experience Replay (HER):

- **HER + Double Q**: the standard configuration using HER and two Q-networks (as in TD3 and SAC).
- **HER + Single Q**: a modified configuration using HER and only one Q-network (i.e., removing the minimum operator in the Bellman target).

To isolate the impact of HER itself, we also include comparisons to versions of each algorithm trained **without HER**. This enables us to quantify the contribution of HER in both simple and complex goal-conditioned tasks.

3.3 Evaluation Metrics

We primarily use **success rate**, **reward**, and **episode length** as our evaluation metrics. All of these are provided directly by the environment.

The success rate is calculated as the percentage of episodes in which the agent successfully achieves the desired goal, defined as:

$$\text{Success Rate} = \frac{\text{Number of Successful Episodes}}{\text{Total Number of Episodes}} \times 100$$

An episode is considered successful if the agent reaches the target goal within a predefined distance threshold (e.g., 5 cm for `PandaReach-v3`). This metric provides a clear and interpretable measure of task completion in goal-conditioned settings.

In our sparse reward formulation, the agent receives a reward of 0 upon success and a reward of -1 otherwise. This binary feedback emphasizes goal completion without providing shaped or dense intermediate signals.

As for episode length, each environment imposes a maximum horizon: 50 steps for `PandaReach-v3` and 100 steps for the more complex `PandaPickAndPlace-v3`. These limits define the upper bound on how long an agent may take to solve each task during an episode.

3.4 Experiment Details

We set the total number of environment interaction steps to 30,000 for `PandaReach-v3` and 2000,000 for `PandaPickAndPlace-v3`, as the latter requires more time to train due to its higher complexity.

The experience replay buffer size is set to 1,000,000 transitions, and the batch size for training is 256. For all algorithms, we use the following hyperparameters unless otherwise specified:

- **Learning rate:** 1×10^{-3}
- **Discount factor** γ: 0.95
- **Policy noise (TD3):** 0.2 (clipped to 0.5)
- **Target smoothing coefficient** τ: 5.0×10^{-3}

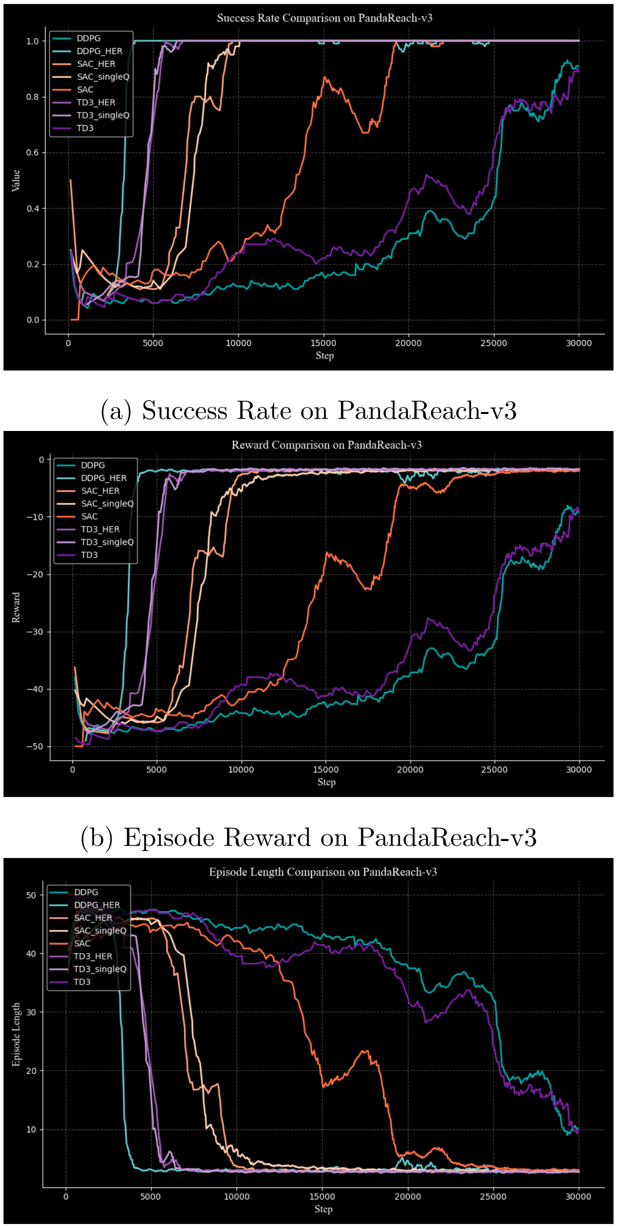
4 Result

4.1 Results on PandaReach-v3

In the `PandaReach-v3` environment with sparse rewards, we trained each algorithm for 30,000 timesteps and evaluated their performance using three key metrics: success rate, episode reward, and episode length. The results are shown in Figures 2(a)–(c).

As shown in Figure 2(a), all algorithms achieved relatively high success rates after 30,000 steps, reflecting the simplicity of the task. However, the convergence speed varied noticeably across methods, revealing the impact of algorithmic design choices such as the use of HER and the Double Q mechanism. Figure 2(b) and Figure 2(c) further illustrate trends in reward and episode length, respectively. Faster convergence is typically associated with higher rewards and shorter episodes, indicating more efficient goal-directed behavior.

From Figures 2(a) and (b), it is clear that TD3 and DDPG converge to around 90% success rate and an average reward of approximately -10, whereas the other algorithm variants approach 100% success rate



(a) Success Rate on PandaReach-v3

(b) Episode Reward on PandaReach-v3

(c) Episode Length on PandaReach-v3

Figure 2: Performance comparison on `PandaReach-v3` across success rate, episode reward, and episode length.

and near-zero reward—indicative of optimal performance. Notably, SAC demonstrates a more stable learning process, achieving convergence earlier (around 20,000 steps), and outperforming TD3 and DDPG in both success rate and final reward.

Effect of Implementing HER

Once Hindsight Experience Replay (HER) is enabled, all three algorithms exhibit significantly improved performance—not only in terms of final success rate, but also in faster convergence and reduced episode length. All HER-enhanced variants achieve approximately -2 to -1 reward and minimal episode length, reflecting efficient and successful goal completion.

These results confirm that HER is a critical augmentation for sparse-reward robotic control tasks. While the overall improvement is consistent with expectations, one surprising observation is that **DDPG + HER** achieves the best final performance among the three HER-augmented algorithms. This is counterintuitive, as DDPG without HER performed the worst. In contrast, the improvement for SAC is relatively limited compared to DDPG and TD3, possibly due to its already strong baseline performance. We will further discuss this behavior and its implications in the next section.

Effect of Removing the Double Q Trick

To validate the impact of removing the Double Q mechanism, we replaced the standard Double Q update ($\min(Q_1, Q_2)$) with a Single Q formulation for both TD3 and SAC. Note that even in the Single Q variant, Hindsight Experience Replay (HER) is still employed to ensure convergence in the sparse-reward setting. DDPG is excluded from this comparison, as it does not implement the Double Q-learning mechanism by design.

When comparing SAC and TD3 with and without the Double Q trick, we observe that TD3 exhibits minimal difference between the two variants. However, in the case of SAC, the Single Q version initially converges faster, suggesting more aggressive value propagation. Yet over time, the Double Q variant eventually surpasses it in final performance, likely due to its more conservative and stable updates.

At this stage, single Q appears to perform poorly compared to double Q, especially in the later stages of training. To further investigate this phenomenon, we extend the analysis to a more complex setting: **PandaPickAndPlace-v3**.

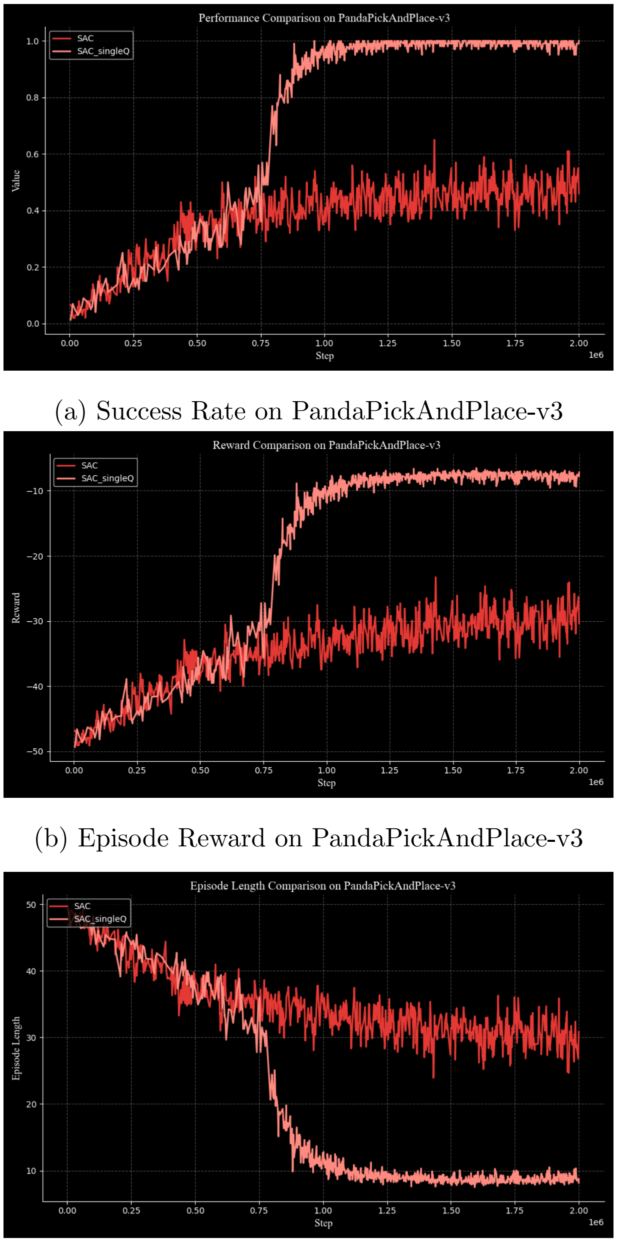
4.2 Results on PandaPickAndPlace-v3

To verify whether the findings from `PandaReach-v3` generalize to more complex environments, we extended our experiments to `PandaPickAndPlace-v3`, a significantly more challenging task. With sparse rewards, we trained each algorithm for 2×10^6 timesteps and evaluated their performance using three key metrics: success rate, episode reward, and episode length, following the same evaluation protocol as in `PandaReach-v3`.

Due to the increased task complexity, we focused on comparing only two variants: **SAC with HER** and **SAC (Single Q) with HER**. The results are shown in Figures 3(a)–(c).

As shown in Figure 3(a), the learning process is considerably slower than in `PandaReach-v3`. Up until around 7.5×10^5 timesteps, both variants perform similarly. However, beyond that point, the Single Q variant shows a surprising acceleration in learning, quickly gaining intuition about the task and converging faster than its Double Q counterpart. In contrast, the Double Q version appears to plateau around a 40% success rate, showing little improvement even with extended training.

A similar trend is observed in Figures 3(b) and (c), which show the episode reward and episode length, respectively. Across all three metrics, Single Q outperforms Double Q after the 7.5×10^5 mark, suggesting that conservative value estimates may hinder progress in more complex goal-conditioned environments, while Single Q enables faster adaptation once early learning stabilizes.



(a) Success Rate on PandaPickAndPlace-v3

(b) Episode Reward on PandaPickAndPlace-v3

(c) Episode Length on PandaPickAndPlace-v3

Figure 3: Performance comparison on `PandaPickAndPlace-v3` across success rate, episode reward, and episode length.

Algorithm	**HER**	**Double Q**	**Success Rate**	**Reward**	**Episode Length**
			PandaReach-v3		
DDPG	No	No	91%	-9.24	10.15
DDPG	Yes	No	**100%**	-1.89	2.89
TD3	No	Yes	89%	-3.50	10.04
TD3	Yes	Yes	**100%**	-9.15	2.73
TD3	Yes	No	**100%**	**-1.66**	**2.67**
SAC	No	Yes	**100%**	-1.83	2.83
SAC	Yes	Yes	**100%**	-1.85	2.85
SAC	Yes	No	**100%**	-1.89	2.89
			PandaPickAndPlace-v3		
SAC	Yes	Yes	46%	-30.40	30.86
SAC	Yes	No	**100%**	**-7.15**	**8.15**

Table 1: Performance comparison across different algorithms and configurations on `PandaReach-v3` and `PandaPickAndPlace-v3`. All metrics are computed over the final evaluation phase: 30,000 timesteps for `PandaReach-v3` and 2×10^6 timesteps for `PandaPickAndPlace-v3`. Rewards and episode lengths are shown with two decimal precision.

5 Discussion

Table 1 summarizes the final performance of all evaluated algorithms across both environments. All metrics were collected at the final stage of training—30,000 timesteps for `PandaReach-v3`, and 2×10^6 timesteps for `PandaPickAndPlace-v3`. The reward and episode length values are reported with two-decimal precision to reflect subtle differences in performance.

5.1 Metric Interpretation and Observations

Although we report three evaluation metrics—success rate, episode reward, and episode length—they typically follow the same trend across all experiments. Higher rewards generally correspond to higher success rates and shorter episode lengths. However, this correlation is not guaranteed in every case.

For instance, even when the success rate reaches 1.0 in `PandaReach-v3`, the corresponding reward and episode length are not necessarily zero. This is expected, as the agent still requires a number of steps to reach the target, and those time steps accumulate a small negative reward due to the sparse reward setting.

Additionally, it is important to interpret the results of `PandaPickAndPlace-v3` carefully. In Figure 3, SAC (Single Q) achieves a success rate close to 1.0, yet it shows slightly lower rewards and longer episode lengths compared to the same algorithm in the `PandaReach-v3` environment. This discrepancy arises from the increased complexity of the task, which involves multiple sub-actions—specifically, grasping the object and placing it at the target location. These actions naturally require more time steps to complete.

As a result, a high success rate in such environments does not necessarily imply optimal efficiency in terms of reward or episode length. In sparse-reward settings, where agents receive limited feedback, the temporal cost of completing multi-step tasks must also be considered when interpreting the metrics.

5.2 Dose HER improve learning in sparse reward problems?

As expected, the use of Hindsight Experience Replay (HER)—which enriches the replay buffer with relabeled goals—enabled agents to extract meaningful learning signals even from otherwise failed episodes.

Interestingly, while SAC is generally considered a stronger baseline compared to DDPG and TD3, our results show that both DDPG + HER and TD3 + HER outperform SAC + HER on `PandaReach-v3`. This trend is consistent with findings in recent work [10], where TD3 + HER achieved the best performance among several off-policy algorithms. Although the cited study did not explicitly explain why SAC + HER underperformed relative to TD3 + HER, we hypothesize that this behavior arises from the fundamental difference in policy types: DDPG and TD3 utilize deterministic policies, while SAC employs a stochastic policy.

In the context of HER, deterministic policies may be better suited to exploit the additional signals provided by relabeled successful outcomes. They can more directly reinforce the beneficial trajectories reconstructed from failed episodes. In contrast, the stochastic nature of SAC may introduce variance that slows the learning process when leveraging those relabeled signals. While SAC's entropy-based exploration encourages diversity and robustness, it may also mitigate the effect of HER's goal relabeling in the early stages of training.

This trade-off between deterministic and stochastic policies under HER presents a compelling direction for future research, particularly in designing hybrid or adaptive strategies that better integrate HER with stochastic policy learning.

5.3 Are Double Q trick always helpful — or sometimes not ?

The Double Q-learning mechanism was originally introduced to address overestimation bias in single Q-learning methods, which often produce overly optimistic Q-values and result in unstable training dynamics. By decoupling action selection and evaluation, Double Q-learning provides a more conservative estimate, leading to more stable learning and improved performance in many settings However, this same conservativeness can become a limitation. In recent years, underestimation has emerged as a potential drawback of Double Q-learning, especially in complex or sparse-reward environments. When Q-values are persistently underestimated, agents may become overly cautious, leading to slower learning or convergence to suboptimal policies.[4]

Our experimental results illustrate both the strengths and limitations of Double Q-learning. In `PandaReach-v3`, Double Q-based methods (e.g., TD3 and SAC with Double Q) benefit from this robustness and ultimately converge quickly and reliably. However, in the more challenging `PandaPickAndPlace-v3` environment, the same conservativeness appears to hinder learning: SAC with Double Q plateaus at around 40% success rate, while the Single Q variant continues to improve and eventually reaches 100% success.

These two cases suggest that the Double Q trick is not universally beneficial. Its effectiveness appears to depend on the task complexity and the required exploration dynamics. While it provides valuable stability in simpler environments, it may become overly restrictive in more complex settings.

That said, it would be premature to conclude that Double Q-learning is unsuitable for all complicated tasks. Further investigation is needed to identify under what specific conditions it degrades learning performance. Recent work has proposed more optimistic variants of Double Q-learning to address this issue [3, 2].Additionally, hybrid strategies that balance or dynamically switch between Single Q and Double Q—such as gradually transitioning from Single Q to Double Q during training—could offer promising avenues for future research.

In summary, our experiments confirm that Double Q-learning, while generally robust, may not always be the optimal choice. We highlight a case where the Single Q formulation outperforms Double Q, encouraging further study into adaptive value estimation strategies tailored to task complexity.

6 Conclusion

This study evaluated the performance of three off-policy actor-critic algorithms—DDPG, TD3, and SAC—on goal-conditioned robotic tasks using the Panda-Gym benchmark, with a focus on the roles of Hindsight Experience Replay (HER) and the Double Q-learning mechanism.

Our results demonstrate that HER is essential for effective learning in sparse-reward environments, particularly for algorithms with deterministic policies such as DDPG and TD3. Furthermore, we find that removing the Double Q trick—thereby relying on a Single Q-network—can accelerate convergence and improve success rates, especially when combined with HER. In this setting, the more aggressive learning behavior of Single Q appears advantageous, and this benefit generalizes across both simple and complex tasks. These observations suggest that conservative value estimation may actually hinder learning when HER already addresses the issue of overestimation.

These findings open the door to new strategies for value estimation in HER-based reinforcement learning. Future research may explore adaptive switching mechanisms between Single and Double Q-learning, or more targeted HER goal relabeling strategies that dynamically respond to task complexity and learning progress.

7 Limitation

Due to time constraints, our experiments were limited to two environments: `PandaReach-v3` and `PandaPickAndPlace-v3`. While these tasks provide valuable insights into goal-conditioned reinforcement learning, further evaluation across a broader range of environments is necessary to fully generalize our findings.

Additionally, each experiment was conducted with a single training run. This increases the likelihood of stochastic variability influencing the results. To improve robustness and statistical reliability, future studies should incorporate multiple random seeds and report averaged performance along with standard deviations. This would not only reduce noise due to randomness but also better define performance boundaries across different configurations.

8 Acknowledgments

We would like to express our gratitude for the assistance and advice provided by Prof. Robert Platt.

References

[1] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay, 2018.

[2] Jacob Buckman, Danijar Hafner, George Tucker, Eugene Brevdo, and Honglak Lee. Sample-efficient reinforcement learning with stochastic ensemble value expansion, 2019.

[3] Kamil Ciosek, Quan Vuong, Robert Loftin, and Katja Hofmann. Better exploration with optimistic actor-critic, 2019.

[4] Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. Diagnosing bottlenecks in deep q-learning algorithms, 2019.

[5] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods, 2018.

[6] Quentin Gallouédec, Nicolas Cazin, Emmanuel Dellandréa, and Liming Chen. panda-gym: Open-source goal-conditioned environments for robotic learning, 2021.

[7] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates, 2016.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[9] Hado Hasselt. Double q-learning. *Advances in neural information processing systems*, 23, 2010.

[10] Xiangkun He and Chen Lv. Robotic control in adversarial and sparse reward environments: A robust goal-conditioned reinforcement learning approach. *IEEE Transactions on Artificial Intelligence*, 5(1):244–253, 2024.

[11] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning, 2019.

[12] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287. Citeseer, 1999.

[13] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, Vikash Kumar, and Wojciech Zaremba. Multi-goal reinforcement learning: Challenging robotics environments and request for research, 2018.

[14] Zhizhou Ren, Guangxiang Zhu, Hao Hu, Beining Han, Jianglun Chen, and Chongjie Zhang. On the estimation bias in double q-learning, 2022.

[15] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 1999.

[16] Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning, 2015.