# Learning Pick&Place Task

**Thabsheer Machingal**
thabsheerjm1@gmail.com

## Abstract

The pick-and-place task remains a benchmark for evaluating a robot's ability to manipulate objects with precision and efficiency. The overall objective of this project is to study the mechanisms and complexities involved in the task and how different learning approaches, including model-free and model-based reinforcement learning (RL), as well as imitation learning, perform. We analyze the performance of these methods based on return over time and training stability. Furthermore, as task generalization remains an open problem, we systematically investigate the robustness of each method against environmental variations. In the current study, we present a trained Soft Actor-Critic (SAC) policy capable of performing the pick-and-place task. We describe the method and reward function used for training and provide an evaluation of the policy's success rate and mean reward acquired.

## 1 Introduction

Pick-and-place serves as a standard benchmark in robotic manipulation due to its simple definition and the richness of challenges it presents, such as precise control, stable grasping, and object relocation under varying conditions. Sample efficiency and training stability are critical concerns for reinforcement learning (RL) in high-dimensional continuous control tasks. Model-free RL techniques, such as Soft Actor-Critic (SAC), can learn complex control policies directly from interaction data but often require extensive training. In this work, we describe our experimental setup and introduce learning policies, including SAC, for a simple pick-and-place task, with a focus on training stability and task performance. Specifically, we present a trained SAC policy and demonstrate its ability to successfully execute the task.

## 2 Related Work

Robotic pick-and-place has long served as a benchmark for examining manipulation capabilities. Recent review highlights the range of challenges involved, including grasping, planning, and generalizing to new objects and environments[1]. Model free methods such as SAC demonstrate strong performance in continuous control tasks including pick-and-place[2]. However, sample efficiency remains a major concern, especially in high-dimensional inputs like images, and various approaches have been proposed to address this limitation[3].

## 3 Methodology

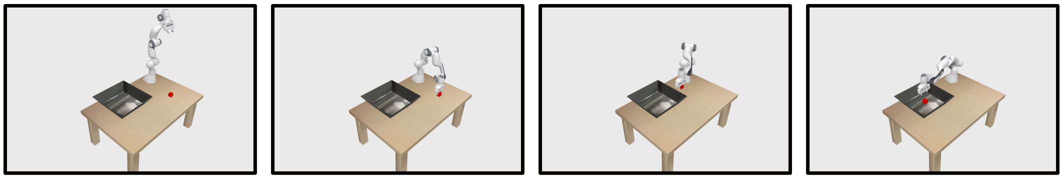
A standard Markov Decision Process (MDP) is defined by the tuple, $\mathcal{M} = (S, A, P, r, \gamma)$. The agent select an action $a \in A$ in each state $s \in S$. A policy P is defined as the probability of selecting an action given the state, $\pi = (a \mid s)$, and our goal is to find an optimal policy, $\pi^* = (a \mid s)$.

For pick-and-place, state $s_t \in S$ can include:

$$s_t = \{q_t, \dot{q}_t, x_{ee,t}, x_{obj,t}, img_t, forces_t\} \tag{1}$$

$q_t, \dot{q}_t$ are the joint positions and velocities respectively, $x_{ee,t}$ is the end-effector pose and $x_{obj,t}$ is the object pose. It is common now to use vision and force features to improve performance of contact-rich tasks.

Preprint. Under review.



(a) Initial State (b) Approach & Grasp (c) In Transit (d) Success

Figure 1: Sequence of the Panda performing pick-and-place task.

Action $a_t \in A$ can be joint commands, torques, or cartesian motions of the end-effector:

$$a_t = \{\Delta q_t, \tau_t, \Delta x_{ee,t}\} \tag{2}$$

However, to simplify and present our setup, in this example we used task space control. our state space contains $x_{ee,t}, x_{obj,t}, gripperstate(g_t)$ and action space has $\Delta q_t$. This simplification reduced dimensionality of the MDP and enabled faster policy search.

To guide the agent in learning the pick-and-place task efficiently, we define a shaped reward function that provides intermediate feedback towards success.

$$r(s_t, a_t) = \begin{cases} r_{\text{reach}}(s_t) & \text{if the object is not grasped} \\ r_{\text{grasp}}(s_t) + r_{\text{place}}(s_t) & \text{if the object is grasped but not yet at the goal} \\ r_{\text{goal}} & \text{if the object is placed at the goal} \end{cases} \tag{3}$$

where

$$\begin{aligned} r_{\text{reach}}(s_t) &= \gamma_1\, f\big(\|x_{ee,t} - x_{obj,t}\|\big), \\ r_{\text{grasp}}(s_t) &= \gamma_2\, \mathbf{1}_{\text{grasped}}, \\ r_{\text{place}}(s_t) &= \gamma_3\, f\big(\|x_{obj,t} - x_{\text{goal}}\|\big), \\ r_{\text{goal}} &= \gamma_4, \\ r_{\text{control}}(a_t) &= -\gamma_5\, \|a_t\|^2. \end{aligned}$$

Here, $x_{ee,t}$, $x_{obj,t}$, and $x_{\text{goal}}$ denote the end-effector, object, and goal positions, respectively. $\mathbf{1}_{\text{grasped}}$ is an indicator function equal to 1 when the object is grasped. The function $f(\cdot)$ encodes a distance-based shaping term ($\tanh$ ), and $\gamma_i$ are hyperparameters controlling the relative contribution of each component. The total reward is summed with a control penalty: $r_{\text{total}} = r(s_t, a_t) + r_{\text{control}}(a_t)$.

In this work we have empirically studied the reward function and tuned the hyperparameters, altogether the shaped reward encourages grasping, moving towards the goal and placing the object while penalizing inefficient motion.

The goal is to find a policy $\pi_\theta(a_t \mid s_t)$ that maximizes the expected return:

$$\pi^* = \arg\max_\theta\, \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{T} \gamma^t\, r(s_t, a_t)\right] \tag{4}$$

### 3.1 Soft-Actor Critic (SAC)

SAC is a model-free, off-policy reinforcement learning algorithm that utilizes the maximum entropy principle to balance the exploration–exploitation trade-off. Unlike other policy gradient methods, SAC explicitly accounts for the entropy of the policy while optimizing it. The actor network learns the policy $\pi_\theta(a \mid s)$, which outputs a probability distribution over actions given a state. The critic network estimates the soft Q-value $Q(s, a)$, which incorporates both the expected return and the policy entropy associated with taking a particular action in a given state.

SAC optimizes a maximum-entropy objective (see eqn (5)) to encourage efficient exploration while maintaining high task performance [2].

$$J(\pi) = \sum_t \mathbb{E}_{(s_t, a_t)\sim\rho_\pi}\left[r(s_t, a_t) + \alpha\mathcal{H}\big(\pi(\cdot \mid s_t)\big)\right] \tag{5}$$

Here, $r(s, a)$ denotes the reward at time step $t$, $\mathcal{H}\big(\pi(\cdot \mid s_t)\big)$ is the entropy of the policy at state $s_t$, and $\alpha$ controls the trade-off between reward maximization and exploration.

## 4 Results and Discussion

The performance of the SAC agent on the pick-and-place task is illustrated in Fig. (2), showing Mean Episode Reward and Success Rate over 1 million timesteps.

The SAC agent demonstrates a three-stage learning trajectory. For the first $10^5$ timesteps, the success rate remains near zero as the agent explores the state space. A significant breakthrough occurs around 400,000 timesteps, where success rate climbs to approximately 50%. Mastery is achieved around 800,000 timesteps, with the agent reaching a stable plateau. At this stage, the Mean Reward stabilizes at approximately 1,000, corresponding to a consistent 100% success rate. Furthermore, the narrowing of the shaded confidence intervals toward the end of training indicates that the policy became highly robust, exhibiting minimal variance across final evaluation episodes.

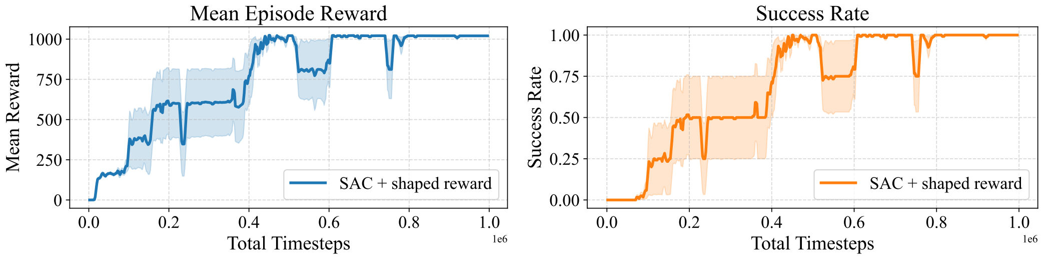


Figure 2: Reward and Success over timesteps (5 seeds): (left) Mean Episode Reward, (right) Success Rate.

The results confirm that the combination of SAC and a shaped reward function is highly effective for solving multi-step dependencies of a pick-and-place task(reaching, grasping, and placing). The "stepped" progression of the learning curve suggests that the agent acquires skills in a modular fashion. The intermittent perfromance drops (e.g., at 230k and 500k steps) are due to SAC's stochastic nature, these depicts the periods where the entropy maximization objective encourages exploration that temporarily degrade the policy. However, the rapid recovery to a 100% success demonstrate the algorithm's sample efficiency and the effectiveness of the shaped reward function in guiding the agent toward the final goal.

## 5 Future Work

This study establishes a foundational baseline for the SAC algorithm on a single-object pick-and-place task in Simulation. Building on these results, we are currently integrating Twin Delayed DDPG (TD3) algorithm to provide a comparative analysis of stability and performance in our next iteration. Our objective is to determine which methodologies best minimize interaction time as we eventually transition toward more complex, multi-object robotic scenarios.

## References

[1] Andrew Lobbezoo, Yanjun Qian, and Hyock-Ju Kwon. Reinforcement learning for pick and place operations in robotics: A survey. *Robotics*, 10(3), 2021.

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

[3] Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. *arXiv:1910.01741*, 2019.