# Evaluating Reward Shaping Strategies in Robotic Learning Tasks

Ioannis Grigoriadis
Dora Medgyesy
Saahith Shetty
Department of Computer Science
Vrije Universiteit Amsterdam
Amsterdam, the Netherlands

August 30, 2025

**Abstract**

This paper investigates the impact of different reward shaping strategies on reinforcement learning performance in robotics tasks. Three reward functions, based on ground truths, observation-driven feedback, and state-machine logic, were evaluated across tasks of increasing complexity. A Deep Q-Network was trained to guide a simulated robot through obstacle avoidance, object collection, and delivery scenarios. Results indicate that aligning reward structure with task demands significantly enhances learning. Additionally, ground-truth penalties improve navigation, observation-based shaping offers a balanced approach, and state-machine rewards support success in multi-phase behaviors.

## 1 Introduction

Conventional machine learning (i.e. supervised learning) utilizes large amounts of data engineered specifically to allow for better learning for the model used. During training, the model is provided with the data by an external source as well as the correct answer. This allows the network to learn and recognize patterns in the dataJordan and Mitchell, 2015. Learning machines are defined as systems that can learn from data without being explicitly programmed, which, although not much different from machine learning, holds a big difference. Learning machines use learning as a method of collecting their own learning data, which is also often called robot learning since it is mainly used in robotics Jones, 2014.

In the context of this project a learning machine will be a Robobo automotive robot tasked with solving a variety of problems of different difficulty levels. The tasks and their design will be:

**Obstacle Avoidance:** The goal of this task is to traverse the environment

without colliding with obstacles, border walls or randomly scattered objects.
**Foraging - Collect:** The goal of this task is to traverse the environment and collect (approach and touch) all food objects scattered around randomly. Limited time is used to avoid the collection of food by chance.
**Foraging - take it home:** The goal of this task is to approach the food object and push it to a designated target area using an extension.
For each task the agent is spawned at the center of the environment.

Artificial intelligence-driven machines are increasingly being used in various aspects of life, such as self driving cars or vacuum cleaners. The agricultural field has also experimented with applying various different learning techniques to farming, fertilization, etc. Smith, 2025. This study investigates a proposed application of learning machines to crop detection and harvesting, thereby incorporating the three proposed tasks. Crops would need to be detected, harvested and brought to a destination by the robot. For this task, a reinforcement learning algorithm would be suitable as that way various visual and sensory inputs from the agricultural land can be used for learning Kaelbling et al., 1996.
Consequently this study investigate the effects, that different reward functions in reinforcement learning have, on the agent's performance, and more specifically: **How does different reward shaping, such as the use of harsh penalties or ground-truth-based signals, affect the agent's performance in various tasks?**

The proposed approach is to test on the above-mentioned tasks with different reward shaping. This way, for each task, the performance of the agent will be evaluated using various reward functions.
The above tasks were selected for their varying objective and challenges and are expected to provide valuable insight into the effects the reward functions have on the performance of the agent.

## 2 Related work

The exploration and analysis of various reward functions for different tasks is a common theme of research. Feng et al., 2024; Nilaksh et al., 2024 evaluate an adaptive reward shaping method that uses Linear Temporal Logic. They dynamically adjust the rewards during training based on the agent's capability to achieve various goals. This type of learning may provide insights into optimizing reward functions used in this study. Another study by Ibrahim et al VaPeP, 2024 highlights the importance of carefully designing a reward function in reinforcement learning. Techniques to create reward functions are discussed such as hierarchical rewards and sub-goal rewards. Reward shaping is also explored, and techniques for modifying rewards during training are given. The results of this paper is important for this study as insights are given into creating an optimal reward function.

# 3 Methodology

The initial step in solving the tasks is creating an optimal reward functions. A deep Q-network is trained to predict the actions of the agent based on the states.

## 3.1 Deep Q Network

Deep Q-learning is used for the completion of each task as it is efficient at handling the high-dimensional observation space created by the sensors and camera input. This method is also model-free, hence the agent can learn effective behaviors directly from rewards without requiring an explicit model of the environment Fan et al., 2020. A Neural Network is used to approximate the Q-function.

Various tasks require different types and amounts of observations. Task 1 uses the 8 IR sensors due to their optimality and simplicity for object detection. Tasks 2 and 3 benefit from using the camera since they have to differentiate between different colored targets. Consequently for task 1 the inputs are the 8 IR sensor data values ($[0, \infty)$) and the per-motor action of the last action $(-100, 100)$. Task 2 has the same inputs and additionally it takes the values of green in each vertical third of the the camera frame $[0, 100]$ as input. Task 3 takes as input all the inputs from task 2 as well as whether red can be found on the bottom horizontal third of the screen ( $B = true, false$).

The action space for Task 1 and 2 are the same, namely: forward, forward_right, forward_left, left, slight_left, right, right_right and backwards. In task 3 the possible actions are forward, left and right due to the negative impacts of sharp turns on the pushing of the object.

During training, the IR senor values and camera information are used as the state space. At each time step the DQN takes this as an input and outputs Q-values, which represent the predicted future reward for taking each possible action in the state. The states, actions, rewards, next states and the done flags are stored in a replay buffer. Mini-batches are sampled from this replay buffer to compute the predicted and target Q-values, and the difference between them is minimized by Smooth L1 loss. The network weights are then updated via backpropagation with the Adam optimiser.

## 3.2 Network Architecture

The network architecture chosen for this research varies slightly depending on the task. The inputs of the network are mentioned above. Each network contains 2 hidden layers of 128 neurons each. The outputs are all the possible actions of each task.

## 3.3 The controller representation

The controller is a trained DQN which acts as the policy for the robot. The inputs of the networks are the IR sensors in task 1 and the IR sensors and the camera information in task 2 and 3. Based on these inputs, the network outputs the Q-values for each action the robot can take in that state. The action with the highest predicted Q-value is chosen and converted to the corresponding motor command of the robot. This allows the robot to move in a given direction to complete the task.

# 4 Experimental set-up

## 4.1 Reward Functions

In the experiments, three reward functions are implemented, where each reward function is tailored to one of the tasks but is applied to each of the tasks during experimentation.

**Function 1** The first reward function is based on the idea that using ground truths can increase the agent's ability to infer information that it would otherwise not know. The agents is given a reward for displacement and continuous 'forward' actions. There is a penalty for crashing into obstacles or a wall, going in circles and repeatedly moving back and forth.

**Function 2** The second reward function uses only knowledge the agent has at its disposal. The only additional step is processing the camera input using a mask. For this function rewards are given for food collection and moving forward. The penalties are given for moving in circles and crashing into an obstacle or wall.

**Function 3** The third reward function uses the idea of a finite state machine where the agent is rewarded differently while in different states Mo et al., 2022. There are four phases, in which the agent receives rewards based on different functions. In phase 1, the agent receives a reward as the target object becomes centered in the field of view and a penalty for oscillating movements. Once the target is centered, the agent moves to phase 2, where a reward is given for moving forward and reducing the distance to the object. When the object reaches a predefined position in the agent's sensor frame, the agent transitions to Phase 3. Here the agent continues to receive location-based rewards while keeping the object in the gripper. Curved or inefficient movements are penalized. Once the goal is detected, the process advances to Phase 4. The agent receives distance-based and visual feedback rewards as it navigates toward the goal. Rewards increase with progress, particularly as the goal becomes more prominent in the sensor input.

A fallback mechanism is implemented, and if the object is lost during Phases 3 or 4, the agent returns to Phase 1 to reacquire it. If the goal is lost during Phase 4, the agent returns to Phase 3 to reorient.

In all experiments, the agent is given an exploration period of 20 episodes. All three reward functions are tested on all three tasks. Additionally, all rewards are scaled for easier comparisons.

When training the agent, in initial steps, there is a higher exploration rate, and the agent chooses random actions more often. As the training progressed, more exploitation was done, and previously known good actions were used.

# 5 Results

### Task 1: Obstacle avoidance

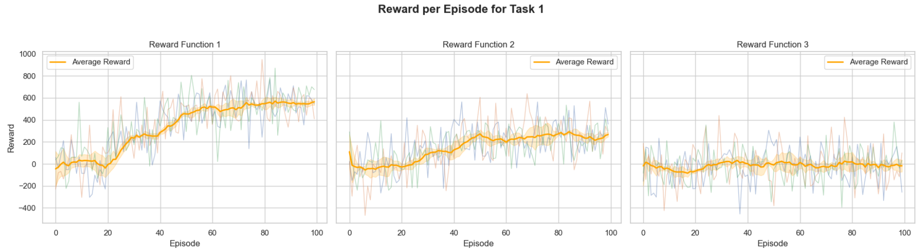

Figure 1: Results for each reward function for Task 1

Figure 1 shows that Reward Function 1 outperforms the others by a significant margin in peak performance, but its large oscillations suggest inconsistent results during training. Reward Function 2 shows some improvement over time, but its general-purpose shaping does not specialize in navigation safety. Reward Function 3 performed the worst, as it is structured around object interaction and transitions that are not optimal for this task.

### Task 2: Foraging, collect

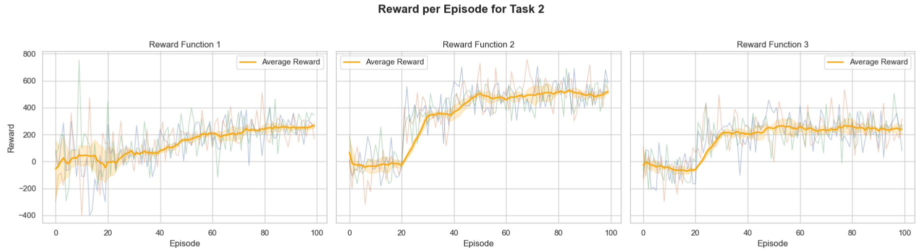

Figure 2: Results for each reward function for Task 2

Here, the agent must collect all food blocks scattered across the environment within a limited time. Figure 2 shows that Reward Function 2 leads to the best performance, enabling the agent to learn an effective collection strategy. Reward Function 1 shows moderate success, likely due to its encouragement of consistent movement, but it lacks a deeper understanding of food, like reward function 2 and 3. Reward Function 3 again under-performs in this task, possibly due to its multi-phase logic being too slow for the time limit. This confirms that reward functions targeting the specific goal behavior are more efficient in shaping performance.

**Task 3: Foraging, take it home.** In this task, the agent must locate a food block, grab it, and transport it to a drop-off zone. Figure 3 shows that

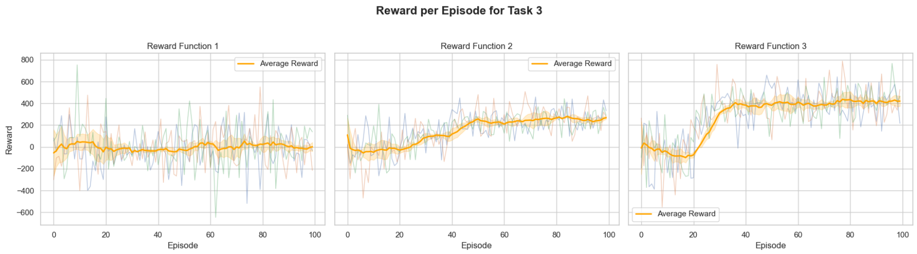

Figure 3: Results for each reward function for Task 3

Reward Function 3 significantly outperforms the others, confirming the advantages of state-based shaping in complex tasksXiao et al., 2023. Its structured phase transitions provide more targeted feedback, leading to stable and consistent learning across runs. Reward Function 2 performs poorly here due to its oversimplified shaping. Reward Function 1 shows nearly random performance, failing to guide the agent through the full behavior chain. These results highlight the importance of structured reward shaping for multi-step or hierarchical goals.

### 5.1 Discussion

Reward Function 1 uses ground-truth data and harsh penalties to enforce structured behavior. While this led to strong performance in obstacle avoidance, it lacked flexibility in more complex, multi-stage tasks. Reward Function 2 relies mainly on observable inputs and applies mild penalties, making it well-suited for time-sensitive tasks like food collection. Its simplicity, however, limited its effectiveness in tasks requiring long-term planning. The state machine used by Reward Function 3 proved highly effective for complex tasks like gathering and pushing food. Conversely, its slower progression and overhead made it less suitable for fast or reactive scenarios.

Due to limited time and computational resources, the scope of more task-optimised reward functions and the number of experimental runs were restricted.

## 6 Conclusions

This work investigated the impact of different reward shaping strategies on the performance of a reinforcement learning agent across three robotic tasks. The reward functions varied in structure, ranging from ground-truth-based to state-machine-driven shaping. Agent behaviour was evaluated using a Deep Q-Network in tasks involving obstacle avoidance, object collection, and delivery.

The results emphasise the need for reward functions that are tailored to the specific structure and requirements of each task. Unlike traditional machine learning, learning machines depend on real-time interaction with their environment, making the design of rewards and observations essential for guiding behaviour. Particularly in complex, multi-stage scenarios.

# References

Fan, J., Wang, Z., Xie, Y., & Yang, Z. (2020). A theoretical analysis of deep q-learning. *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, *120*, 486–489.

Feng, L., Elsayed-Aly, I., & Kwon, M. (2024). Adaptive reward design for reinforcement learning in complex robotic tasks. *ArXiv*, *abs/2412.10917*. https://doi.org/10.48550/arXiv.2412.10917

Jones, N. (2014). Computer science: The learning machines. *Nature*, *505*, 146–148. https://doi.org/10.1038/505146a

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255–260. https://doi.org/10.1126/science.aaa8415

Kaelbling, L. P., Littman, M. L., & Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, *4*, 237–285.

Mo, Y.-W., Ho, C., & King, C.-T. (2022). Managing shaping complexity in reinforcement learning with state machines-using robotic tasks with unspecified repetition as an example. *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, 544–550.

Nilaksh, N., Ranjan, A., Agrawal, S., Jain, A., Jagtap, P., & Kolathaya, S. N. Y. (2024). Barrier functions inspired reward shaping for reinforcement learning. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 10807–10813. https://doi.org/10.1109/ICRA57147.2024.10610391

Smith, M. S. (2025). Agricultural weed control is a delicate process. ai tools could transform how farmers tackle it [Accessed: 2025-06-27]. *Business Insider*. https://www.businessinsider.com/ai-tools-weed-control-efficiency-farming-agriculture-2025-6?international=true&r=US&IR=T

VaPeP, A. (2024). A comprehensive overview of reward engineering and shaping in advancing reinforcement learning applications. *arXiv preprint arXiv:2412.10917*. https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10763475

Xiao, R., Jiang, Y., Yang, C., & Zhan, H. (2023). Demonstration shaped reward machine for robot assembly reinforcement learning tasks, 232–244. https://doi.org/10.1007/978-981-99-6495-6_20