Reward Shaping Techniques in Reinforcement Learning

Ariana Dariuni [1]

[1]Sharif University of Tech / CE department

1. Introduction & Motivation

Reinforcement Learning (RL) trains agents through trial and error, guided by a reward signal. However, in many complex, real-world problems, rewards are extremely **sparse**. An agent might perform thousands of actions before receiving any feedback, leading to inefficient random exploration and often a complete failure to learn.

Reward Shaping is a technique to address this by creating a richer, denser reward signal. By providing frequent "hints," we guide the learning process, dramatically accelerating the agent's ability to solve the task.

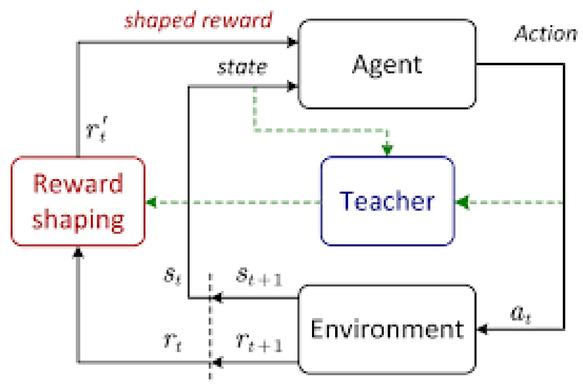

Figure 1. Conceptual diagram of an RL agent receiving sparse vs. shaped rewards.

2. The Problem: Reward Sparsity

- **Definition:** A scenario where meaningful rewards are received only after completing a long sequence of correct actions.
- **Consequence:** The agent cannot distinguish between good and bad actions during exploration, as most actions result in a reward of zero.
- **The Challenge:** How can we provide guidance without altering the optimal behavior or introducing exploits?

3. The Foundation: Potential-Based Reward Shaping (PBRS)

The most common and theoretically-grounded method for reward shaping. It adds an auxiliary reward based on the *change in potential* between states.

The PBRS Formula: The new, shaped reward R' is defined as:

$$R'(s,a,s') = R(s,a,s') + \underbrace{\gamma\Phi(s') - \Phi(s)}_{\text{Shaping Reward}}$$

- $R(s,a,s')$: Original, sparse environment reward.
- $\Phi(s)$: A user-defined **potential function** that estimates the "value" of being in state s.
- γ: The discount factor.

The Theoretical Guarantee: Policy Invariance
PBRS is guaranteed to **preserve the optimal policy**. The agent learns the *same* optimal behavior, but significantly faster.

4. Advanced Methods: A Formal Overview

These methods automate reward design by framing it as an optimization or inference problem.

A. Learning the Reward Function

Problem: Manually specifying rewards is brittle. Instead, we can learn them as part of the training process.

Policy Gradient for Reward Design (PGRD)

- **Objective:** Find reward parameters θ^* that maximize the expected *true* objective reward R_O.

$$\theta^* = \arg\max_\theta \lim_{N\to\infty} \mathbb{E}\left[\frac{1}{N}\sum_{t=0}^{N} R_O(s_t) \mid R(\cdot;\theta)\right]$$

Learning Intrinsic Rewards (LIRPG)

- **Mechanism:** Update policy π_θ using a gradient from combined extrinsic and intrinsic returns G_{ex+in}.

$$\theta' \approx \theta + \alpha G_{ex+in}(s_t,a_t)\nabla_\theta \log \pi_\theta(a_t|s_t)$$

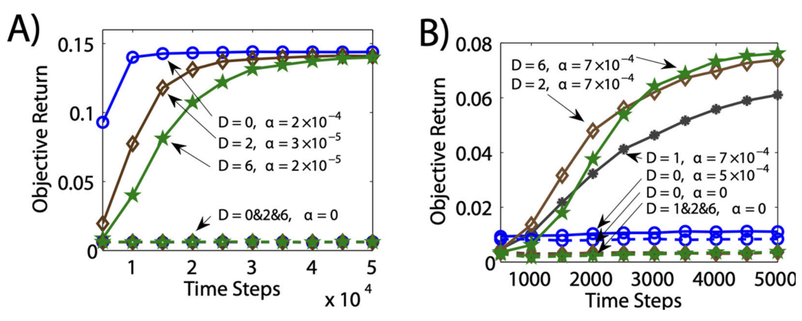

Figure 2. PGRD performance with A) poor model, B partially observable world

B. Shaping for Intelligent Exploration

Problem: In large state spaces, guide exploration systematically by adding a weighted intrinsic reward.

$$R_{total} = R_{extrinsic} + \beta R_{intrinsic}$$

Novelty-Driven (Hash-Based Exploration)

- **Key Idea:** Reward visiting low-count states.
- **State Hashing:** $\phi(s) = \text{sgn}(Ag(s)) \in \{-1,1\}^k$
- **Intrinsic Reward:** $R_{int}(s) = \frac{1}{\sqrt{N(\phi(s))}}$

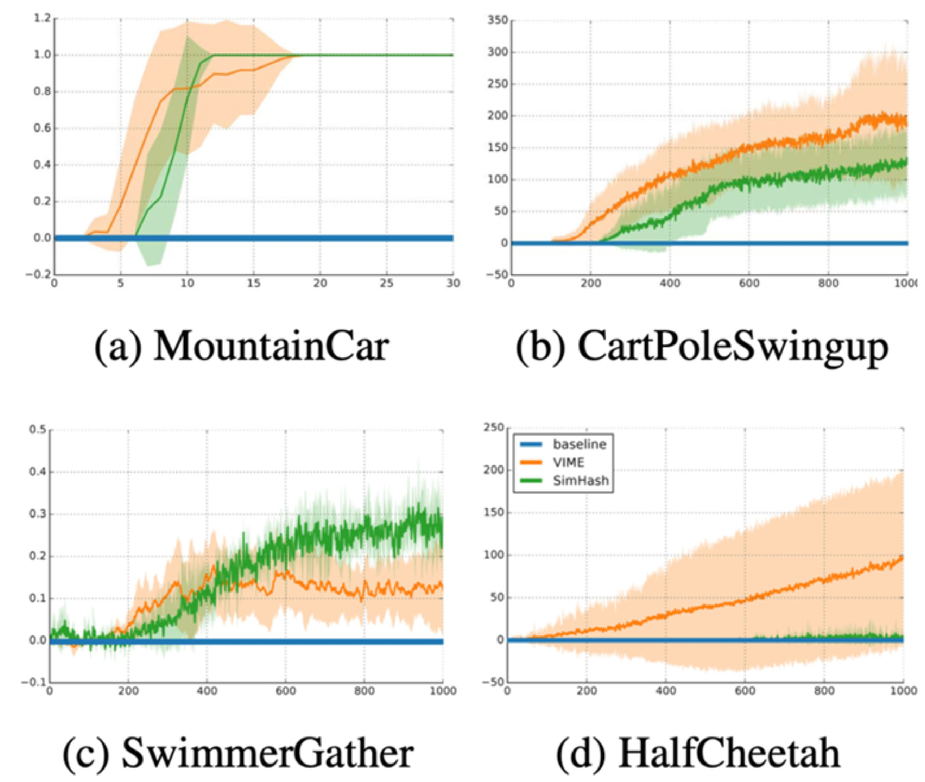

Figure 3. SimHash algorithm successfully solves sparse-reward tasks where the baseline fails, and performs competitively with the VIME algorithm.

B. Shaping for Exploration (Cont.)

Curiosity-Driven (Information Gain - VIME)

- **Key Idea:** Reward reducing uncertainty about the world dynamics model ψ.
- **Intrinsic Reward:** Information gain measured by the KL-Divergence.

$$R_{int}(s_t,a_t) = D_{KL}\left[p(\psi|s,a,s') \parallel p(\psi|s,a)\right]$$

Uncertainty-Driven (Ensemble Methods - RUNE)

- **Key Idea:** Reward exploration where an ensemble of reward models $\{\hat{R}_i\}$ disagrees.
- **Total Reward:** Mean (exploitation) + Std. Dev. (exploration).

$$R_{total} = \mathbb{E}_i\left[\hat{R}_i\right] + \beta \cdot \text{Std}_i\left[\hat{R}_i\right]$$

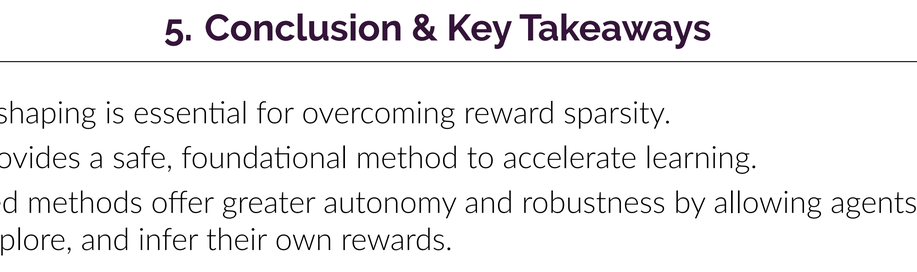

Figure 4. RUNE uses uncertainty in learned reward functions as an exploration bonus.

C. Inferring True Objectives

Problem: Manually designed rewards can be exploited (**reward hacking**). Instead, infer the objective from expert data.

Inverse Reinforcement Learning (IRL)

- **Objective:** Given expert trajectories τ_E, find $\hat{R}$ for which π_E is the optimal policy.

Inverse Reward Design (IRD)

- **Key Idea:** Treat a proxy reward R_{proxy} as noisy evidence of a true reward R_{true}.
- **Bayesian Inference:** $P(R_{true}|R_{proxy}) \propto P(R_{proxy}|R_{true})P(R_{true})$

5. Conclusion & Key Takeaways

- Reward shaping is essential for overcoming reward sparsity.
- PBRS provides a safe, foundational method to accelerate learning.
- Advanced methods offer greater autonomy and robustness by allowing agents to learn, explore, and infer their own rewards.
- The field is moving towards more aligned and autonomous agents capable of solving complex, real-world problems.