Progress Report: Advanced Machine Learning Research Paper

Date	August 26, 2025
Student ID	210362X
Project Code	RL004
Project Title	Hindsight Experience Replay (HER) – PyTorch Implementation and Experiments on Fetch Robotic Environments
Course	CS4681 – Advanced Machine Learning
Instructor	Dr. Uthayasanker Thayasivam

Baseline Model

Hindsight Experience Replay (HER), an off-policy reinforcement learning approach enhancing learning in sparse and binary reward settings by "re-labeling" goals from failed episodes.
Based on: [1707.01495] Hindsight Experience Replay

1. Introduction and Problem Statement

Reinforcement learning (RL) in environments with sparse and binary rewards is notoriously sample-inefficient. The method known as Hindsight Experience Replay (HER), introduced by Andrychowicz et al. (2017), allows agents to learn from failures by retrospectively treating achieved end-states as pseudo-goals, relabeling transitions to produce more learning opportunities [1].

This technique can integrate with any off-policy RL algorithm, significantly improving sample efficiency and making learning feasible in robotic tasks like pushing, sliding, and pick-and-place with only binary success indicators [2].

In our project, we aim to explore and reproduce this approach via a PyTorch implementation (TianhongDai's code repository), and test HER on various Fetch robotics environments.

2. Literature Review

Hindsight Experience Replay (HER)

HER repurposes failed episodes by relabeling goals with those actually achieved later in the trajectory, providing informative feedback even when the original goal is not met. The approach boosts learning effectiveness in sparse-reward domains and has been validated on multiple robotic manipulation tasks. Its core mechanism reframes unsuccessful transitions as successes toward alternate goals [3], [2].

Goal-sampling Strategies in HER

There are several ways to select new pseudo-goals:

- **final**: use the final state of an episode
- **future**: sample a state from later in the episode (commonly k = 4 or 8 yields strong results)
- **episode**: pick any state within the current episode
- **random**: sample from the entire replay buffer [3]

    The "future" strategy with k=4 or 8 generally performs best in practice across robotic manipulation tasks [3].

PyTorch Implementation (TianhongDai repository)

The repository implements HER in PyTorch across all Fetch robotics environments and includes modules like her_modules, rl_modules, argument parsing, and demo scripts [4].

Key features include:

- **Requirements**: Python 3.5.2, OpenAI Gym 0.12.5, Mujoco-py 1.50.1.56, PyTorch 1.0.0, mpi4py [4].
- **Usage examples**: Commands for training environments such as FetchReach-v1, FetchPush-v1, FetchPickAndPlace-v1, and FetchSlide-v1, using MPI for parallel processing [4].
- The her_sampler class in her_modules/her.py encapsulates the sampling strategy and reward logic [4].
- A demo script (demo.py) loads models and visualizes agent performance with options for goal normalization and rendering [4].

3. Proposed Methodology

Baseline Model

We will adopt the standard HER approach as implemented in the PyTorch repository by TianhongDai, using default goal-sampling settings such as the "future" strategy.

Experimental Enhancement (Possible Future Work)

- Investigate variations of goal sampling strategies (e.g., final, episode, random) and replay-k configurations to compare performance.
- Analyze the impact of different HER strategies on convergence rate and success in each Fetch environment.

Implementation Plan

1. Set up the environment with required dependencies.
2. Replicate training runs for environments: FetchReach, FetchPush, FetchPickAndPlace, FetchSlide.
3. Experiment with varying replay_k and strategies (future vs final vs others).
4. Visualize learning curves and compare sample efficiency across settings.

Hypothesis

Using the "future" HER strategy with an optimal replay-k (e.g., 4 or 8) will lead to faster convergence and significantly more efficient learning in Fetch environments, compared to baseline no-HER versions.

4. Experimental Plan

- **Datasets / Environments**: Gym Fetch environments: FetchReach-v1, FetchPush-v1, FetchPickAndPlace-v1, FetchSlide-v1.
- **Evaluation Metrics**:
  - Success rate over episodes (e.g., proportion of successful attempts).
  - Learning efficiency: number of episodes or timesteps required to reach a threshold performance.
  - Impact of different HER strategies (future vs final vs others) and values of k on performance.
- **Experimental Setup**:
  1. **Baseline Replication**: Run the original HER implementation with default settings to reproduce expected performance trends.
  2. **Strategy Comparison**: Run experiments with different goal sampling strategies and k-values.
  3. **Visualization & Analysis**: Plot learning curves (median over seeds), compare convergence speed and final success rates.
- **Expected Outcome**: We anticipate that the default "future" strategy with an appropriate replay_k produces the highest sample efficiency and fastest learning across tasks. Alternative strategies may reveal trade-offs in stability vs. speed.

5. Project Timeline (Gantt Chart)

Task	Weeks 1–2	Weeks 3–4	Weeks 5–6	Weeks 7–8	Weeks 9–10	Weeks 11–12

Setup environment & dependencies	[redacted]					
Baseline reproduction of HER		[redacted]				
Implement variations & run experiments			[redacted]			
Evaluate & visualize results				[redacted]		
Analyze and write report					[redacted]	[redacted]

6. References

1. Andrychowicz, M. et al. (2017). *Hindsight Experience Replay*. arXiv:1707.01495.
2. Reinforcement Learning with Sparse Rewards. *Technische Universität Chemnitz*.
3. Goal-sampling strategies discussion. *GitLab*.
4. TianhongDai. *Hindsight Experience Replay (PyTorch) GitHub repository*.