Hindsight Foresight Relabeling For Meta-Reinforcement Learning

Michael Wan, Jian Peng, Tanmay Gangwani

ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

Motivation

- Prior work in multi-task RL [1, 2] has successfully shared data among tasks through reward relabeling
- In multi-task RL, trajectory τ collected for task ψ^i can be used to learn task ψ^j if return of τ under ψ^j is high

Meta-RL vs Multi-Task RL

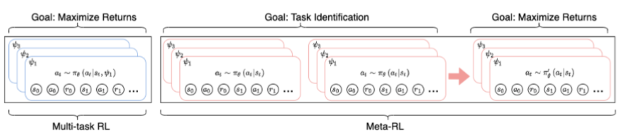
- Multi-Task RL: given task $\psi \sim p(\psi)$, RL agent seeks to maximize its returns

$$max_\theta \mathbb{E}_{\psi \sim p(\psi),\, s_t,\, a_t \sim \pi_\theta}[\textstyle\sum_{t=1}^{\infty} \gamma^{t-1} r_\psi(s_t, a_t)]$$

- Meta-RL: agent must learn to identify task, then maximize returns

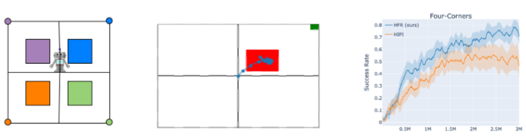
$$max_{\theta,\phi} \mathbb{E}_{\psi \sim p(\psi),\, s_t,\, a_t \sim \pi'(\theta,\phi)}[\textstyle\sum_{t=1}^{\infty} \gamma^{t-1} r_\psi(s_t, a_t)]; \quad \pi'(\theta, \phi) = f_\phi(\pi_\theta, \tau_{pre}, r_\psi)$$

where f_ϕ is the adaptation procedure, $\pi'(\theta, \phi)$ is the post-adaptation policy



Claim: Reward relabeling based on trajectory returns works for multi-task RL but may be sub-optimal for Meta-RL!

An Illustrative Example



Relabeling based on returns is sub-optimal in this environment. HFR avoids this by relabeling based on expected post-adaptation returns.

- Four-Corners Environment
  - Goal locations in each of the four corners correspond to tasks
  - For each goal, there is a section in the corresponding quadrant that gives large negative reward
- Consider a trajectory that hovers over the blue square
  - Clearly useful for task identification for the blue task
  - Highly negative return under the blue task, so relabeling methods based on returns will not assign the trajectory to the blue (correct) task
- **HFR** correctly assigns the trajectory to the blue task

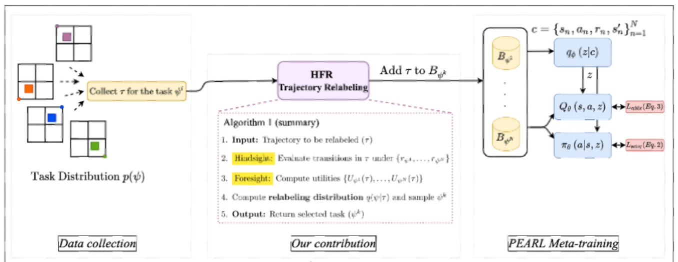
Relabeling for Meta-RL (HFR)

- We use **HFR** to relabel based on *post-adaptation returns* (utility). This aligns with meta-RL objective

$$U_\psi(\tau) = \mathbb{E}_{s_t,\, a_t \sim \pi'(\theta,\phi)}[\textstyle\sum_{t=1}^{\infty} \gamma^{t-1} r_\psi(s_t, a_t)]; \quad \pi'(\theta, \phi) = f_\phi(\pi_\theta, \tau, r_\psi)$$

Hindsight: Relabel trajectory using reward functions for different training tasks
Foresight: Compute expected post-adaptation returns (utility) after using the relabeled trajectory for adaptation for different tasks

HFR can be incorporated into any off-policy meta-RL algorithm; we use PEARL [3]



Relabeling is independent of data collection and update rules of underlying meta-RL algorithm

- Trajectory τ is more likely to be assigned to tasks for which it has higher (normalized) utility. We use the relabeling distribution:

$$q(\psi \mid \tau) \propto p(\psi) e^{U_\psi(\tau) - log Z(\psi)}$$

  - Based on prior work on relabeling in multi-task RL [1]
- Sampling trajectories to compute post-adaptation returns is expensive
  - We approximate utility using Q function:

$$U_\psi(\tau) = \mathbb{E}_{s_1 \sim p(s_1),\, a_1 \sim \pi'(\cdot \mid s_1)}[Q_\psi^{\pi'}(s_1, a_1)]$$

Algorithm 1: Hindsight Foresight Relabeling (HFR)

Input : Trajectory to be relabeled (τ)
Output : Task to relabel the trajectory with (ψ)

for *each training task* ψ^i **do**
  $U_{\psi^i}(\tau) \leftarrow$ `ComputeUtility`(τ, ψ^i)
  $\log Z(\psi^i) \leftarrow$ `GetLogPartition`(ψ^i)
end
Return $\psi \sim \text{softmax}\{U_{\psi^i}(\tau) - \log Z(\psi^i)\}$ (Eq. 8)

Function `GetLogPartition` (ψ) **:**
  Sample batch of trajectories $\{\tau^i\}_{i=1}^N \sim B_\psi$
  for *each trajectory* τ^i **do**
    $U_\psi(\tau^i) \leftarrow$ `ComputeUtility`(τ^i, ψ)
  end
  Return $\log Z(\psi) \approx \log\left(\frac{1}{N}\sum_{i=1}^N e^{U_\psi(\tau^i)}\right)$

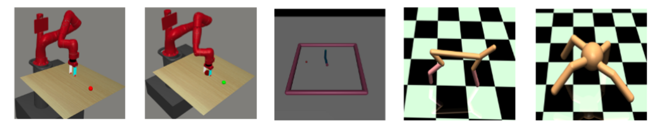
Function `ComputeUtility` (τ, ψ) **:**
  for *each* $(s_t, a_t, r_t) \in \tau$ **do**
    Replace r_t with $r_\psi(s_t, a_t)$
  end
  Sample embedding using encoder $z \sim q_\phi(z|\tau)$
  Sample a batch of initial states $\{s_1^i\}_{i=1}^N \sim B_\psi$

  Sample actions for these states using the post-adaptation policy $\pi_\theta(\cdot|s, z)$:
  $\{a_1 \sim \pi_\theta(a_1^i|s_1^i, z)\}_{i=1}^N$

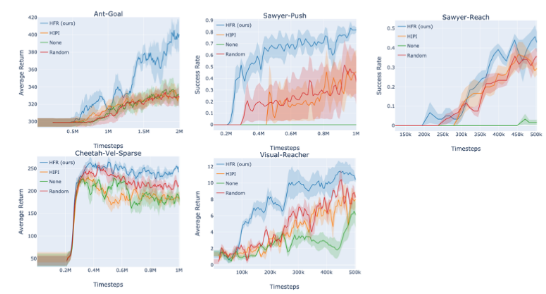
  Return $U_\psi = \frac{1}{N}\sum_{i=1}^N Q_\theta(s_1^i, a_1^i, z)$ (Eq. 9)

Experimental Results

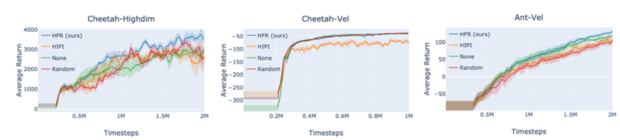
- Evaluate on sparse and dense reward manipulation and locomotion tasks
- Tasks involve low-dimensional state vectors and high-dimensional images (Visual-Reacher) as input
- Methods
  - **HFR (ours)**
  - HIPI [1] (trajectory return value-based relabeling algorithm)
  - Random (assign trajectories to random tasks)
  - None (PEARL with no relabeling)



- **Results: Sparse-Reward Tasks**



- **Results: Dense-Reward Tasks**



Further Analysis in the Paper

- Ablations on batch size and role of the partition function
- HFR with learned reward functions
- Effect of stochasticity in the relabeling distribution

HFR Code

[1] Rewriting History with Inverse RL: Hindsight Inference for Policy Improvement, Eysenbach et al.
[2] Generalized Hindsight for Reinforcement Learning, Li et al.
[3] Efficient Off-Policy Meta-Reinforcement Learning via Probabilistic Context Variables, Rakelly et al.