Reinforcement Unlearning[1]

백승언

14 May, 2026

[1] https://arxiv.org/abs/2312.15910

Contents

- **Introduction**
  - Machine unlearning
- **Reinforcement Unlearning**
  - Motivation
  - Problem formulation
  - Challenges of unlearning in RL
  - Proposed methods
- **Experiments**
  - Experimental settings
  - Results

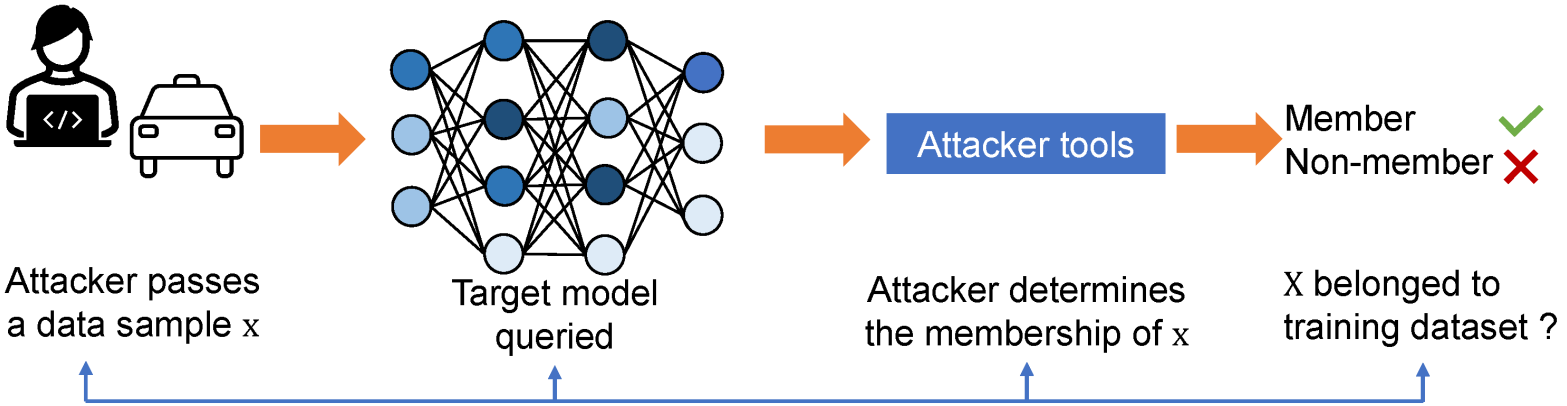
Introduction

Machine Unlearning

- **The process of removing the influence of specific training data from a trained model**
  - Dirven by the “right to forgotten” and the need to erase sensitive or outdated information
  - Formally, an unlearning algorithm R is considered valid if the following condition holds for a given dataset D and target sample $d \in D$, specific learning algorithm A.

$$\mathbb{P}\big(R(A(D), d)\big) \approx \mathbb{P}\big(A(D \backslash \{d\})\big)$$

- **Membership Inference Attack (MIA): the baseline threat**
  - MIA deduces whether a specific individual sample was used in the model’s training set
  - MIA operates at the individual sample level, focusing on discrete data points



Concept of membership inference attack (MIA)

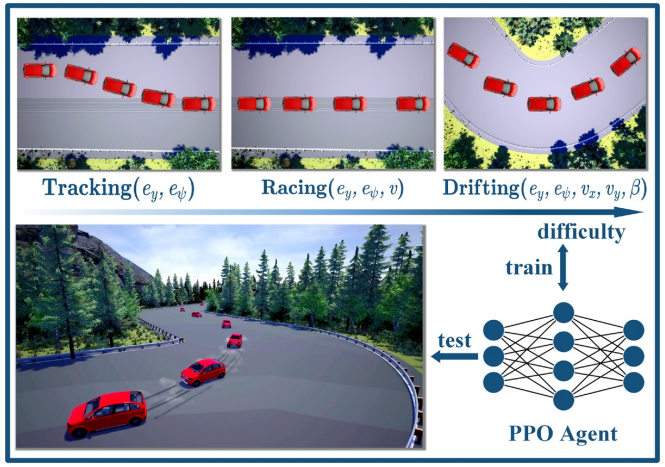
Reinforcement Unlearning

Motivation: Why do we need unlearning in RL?

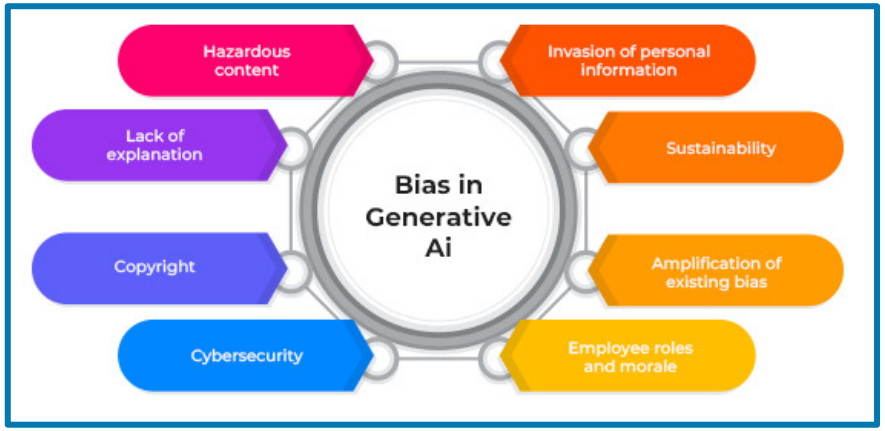
- **Scenario 1: Privacy protection**
  - Recommendation systems retaining sensitive user history after deletion requests
  - Autonomous systems memorizing personal trajectories and location patterns

- **Scenario 2: Rights to be forgotten**
  - GDPR/CCPA requires removing not just raw data, but its influence embedded in model params

- **Scenario 3: Policy correction and safety**
  - Agents trained on poisoned or obsolete environments may persist in harmful strategies
  - A financial trading AI overfitted to a Black Swan event repeatedly making aggressive trades inconsistent with current market conditions
  - An LLM fine-tuned on historically biased conversation set consistently generates stereotyped or discriminatory responses



RL-based autonomous vehicle



Bias in generative AI

Problem formulation

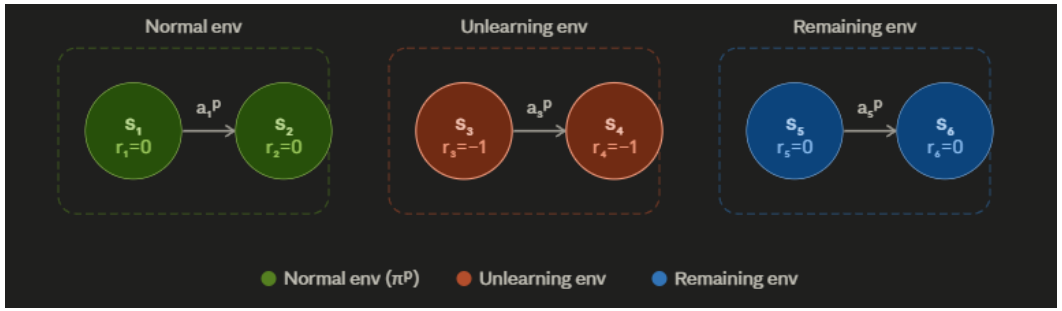
- **The objective of reinforcement unlearning is to eliminate the influence of a specific env on the agent, i.e., "forgetting an environment".**
  - This definition may not align with the conventional machine unlearning approach; this difference arises because reinforcement unlearning operates within a **distinct learning paradigm**.

- **Formally, the authors define the reinforcement unlearning problem as follows,**
  - They consider a set of n learning environments $\{\mathcal{M}_1, \dots, \mathcal{M}_n\}$
    - Each env $\mathcal{M}_i$ has the same state $\mathcal{S}$ and action spaces $\mathcal{A}$ but differs in state transition $\mathcal{P}$ and reward function $\mathcal{R}$.
  - They define the target environment to be unlearned as $\mathcal{M}_u =< \mathcal{S}_u, \mathcal{A}_u, \mathcal{P}_u, \mathcal{R}_u >$.
    - The set of remaining envs, denoted as $\{\mathcal{M}_1, \dots, \mathcal{M}_{u-1}, \mathcal{M}_{u+1}, \dots, \mathcal{M}_n\}$ will be referred to as retaining environments.
  - Given a policy π, **the goal is to update the policy π to π'** such that the accumulated reward obtained in $\mathcal{M}_u$ is minimized, while the accumulated reward received in the retaining envs remains the same.
    - The objective is defined as follows

$$\text{Object 1 (Forgetting Goal)}: \min_{\pi'} \left|\left|Q_{\pi'}(s_u)\right|\right|_{\infty}, \qquad s_u \in \mathcal{S}_u,$$
$$\text{Object 2 (Retaining Goal)}: \min_{\pi'} \left|\left|Q_{\pi'}(s_r) - Q_{\pi}(s_r)\right|\right|, \qquad s_r \notin \mathcal{S}_u.$$

Challenges of unlearning in RL

- **The authors introduce three specific challenges in reinforcement unlearning**
  - Sequential Dependency
    - Unlike i.i.d. data in supervised learning, RL involves **interconnected trajectories within episodes**.
    - Forgetting requires removing "Environment Dynamics" embedded in the policy, **not just a single data point**.
  - Stability-Forgetting Trade-off
    - Conflict between Obj_1 (Forgetting) and Obj_2 (Retaining).
    - Must achieve "**Surgical Removal**" of target memories without triggering **Catastrophic Forgetting** in other envs.
  - Verification Ambiguity
    - Model weights don't explicitly show what has been forgotten.
    - Needs a formal framework like **Environment Inference Attack** (EIA) to prove the information is truly purged.



Sequential dependency in RL

Proposed methods (I) – overview

- **The authors assume the Environment Inference Atack (EIA) situation**
  - Adversary's strategy: adversaries attempt to infer the specific characteristics of a learning env by observing the agent's interaction samples within that environment
    - Adversary's goal: Infer the transition function $\mathcal{P}_u$
    - Available information: adversary can access to $\mathcal{S}_u, \mathcal{A}_u, r_u$ and the unlearned policy π'

- **They propose two distinct unlearning methods: decremental RL, environment poisoning.**
  - The decremental RL method involves updating the agent by minimizing its reward in the unlearning env
    - Concept: reducing agent effectiveness in unlearning env

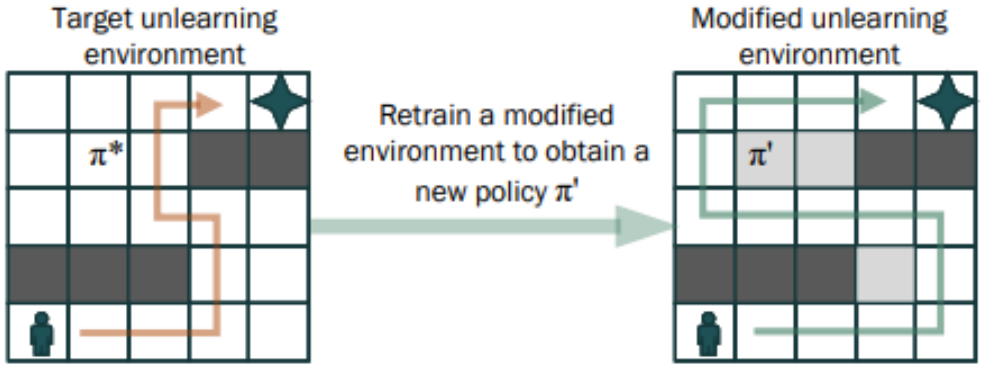
$$\mathcal{L}_u = \mathbb{E}_{s\sim\mathcal{S}_u}\left[\left|\left|Q_{\pi'}(s)\right|\right|_\infty\right] + \mathbb{E}_{s\nsim\mathcal{S}_u}\left[\left|\left|Q_{\pi'}(s) - Q_{\pi}(s)\right|\right|_\infty\right]$$

  - The env poisoning method focuses on modifying the state transition and reward function.
    - Concept: after poisoning, inducing the agent to learn incorrect or sub-optimal behaviors in $\mathcal{M}_{\mathrm{u}}$.

$$R_i \coloneqq \lambda_1 \Delta(\pi_i(s_i)|\pi_{i+1}(s_i)) + \lambda_2 \sum_{s\nsim\tau_u}\sum_{a}\pi_i(s,a)r(s,a),$$

Proposed methods (II) – Environment poisoning-based method

- **The env poisoning method aims to influence the agent's policy learning by creating a situation where its previously learned knowledge becomes less effective**
  - This method consists of three steps
    - 1. Apply a random poisoning strategy to **alter the transition function** of the unlearning env
      - $\hat{\mathcal{P}}_u(s'|s,a) = g(s,a)$
    - 2. Agent learns a new policy in this modified env
    - 3. Based on the agent's learned policy, **they update the poisoning strategy** and re-poison the unlearning env
      - $R_i \coloneqq \lambda_1 \Delta(\pi_i(s_i)|\pi_{i+1}(s_i)) + \lambda_2 \sum_{s \curlywedge \tau_u a \sim \pi} \sum_a \pi_i(s,a) r(s,a)$



Concept of env poisoning-based method

Algorithm 1 The poisoning-based method

Input: The learned policy π^*, the learning tasks $\mathcal{M}_1, ..., \mathcal{M}_n$, and the unlearning task $\mathcal{M}_u$;
Output: A refined policy $\hat{\pi}$;

for poisoning epoch $i = 1, ..., m$ **do**
  Select a poisoning action g_i from $\mathcal{G}$;
  Alter the transition function of the unlearning task from $\hat{\mathcal{T}}_{u,i-1}$ to $\hat{\mathcal{T}}_{u,i}$ based on g_i;
  The agent learns a policy π_i according to $\hat{\mathcal{T}}_{u,i}$;
  Receive reward $\mathcal{R}_i$;
  Update poisoning strategy using samples $(\pi_{i-1}, g_i, \pi_i, \mathcal{R}_i) \in \mathcal{B}$ by optimizing loss function Eq. 2;
end for
return $\hat{\pi} \leftarrow \pi_m$;

Pseudo code of env poisoning-based method

Experiments

Experimental settings (I)

- **Tasks**
  - Goal-based tasks
    - General tasks such as grid world, virtual home, maze explorer
    - Action: up, down, left, right
  - Recommendation task
    - Dataset: MovieLens, Envs: Each user, State: historical ratings per each user: Action: specific movie index, Reward: user's true rating
  - Aircraft landing task
    - Similar with grid world, this task aims to guide an aircraft to its landing goal while avoiding obstacles in the env
- **Metrics**
  - Cumulative reward
  - The number of steps in goal-based tasks
  - Recommendation accuracy in recommendation task
  - Collision count in aircraft landing task

Experimental settings (II)

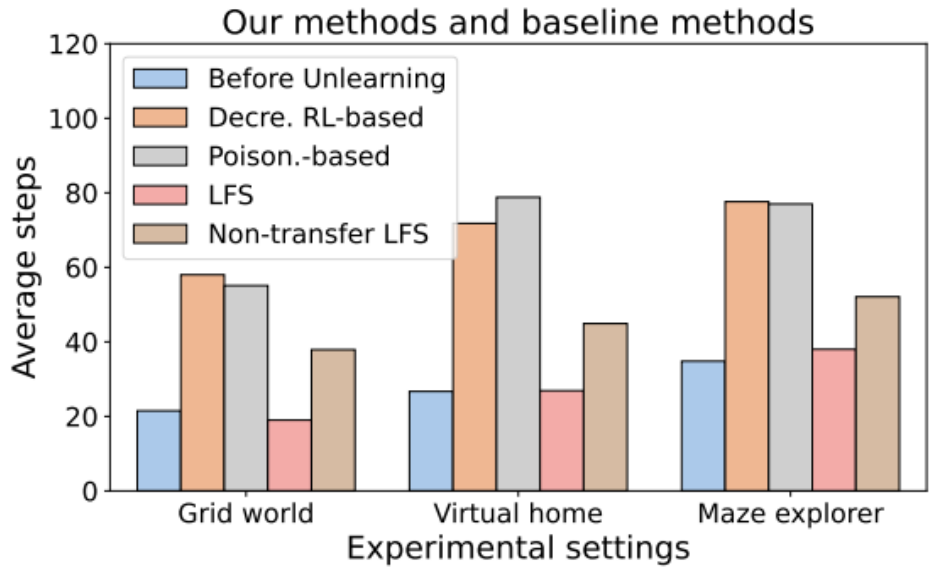
- **Baselines**
  - Learning from scratch (LFS)
    - Removing unlearning environment and then retraining the agent from scratch using the remaining envs
  - Non-transferable learning from scratch (Non-transfer LFS)
    - Labeling the experience samples from all learning environments and then training the agent with following dual loss

$$\mathcal{L} = \begin{cases} -\mathbb{E}\left[\left(r(s,a) + \gamma \max_{a'} Q(s',a';\theta) - Q(s,a;\theta)\right)^2\right] & if\ s \in \mathcal{S}_u \\ \mathbb{E}\left[\left(r(s,a) + \gamma \max_{a'} Q(s',a';\theta) - Q(s,a;\theta)\right)^2\right] & otherwise \end{cases},$$

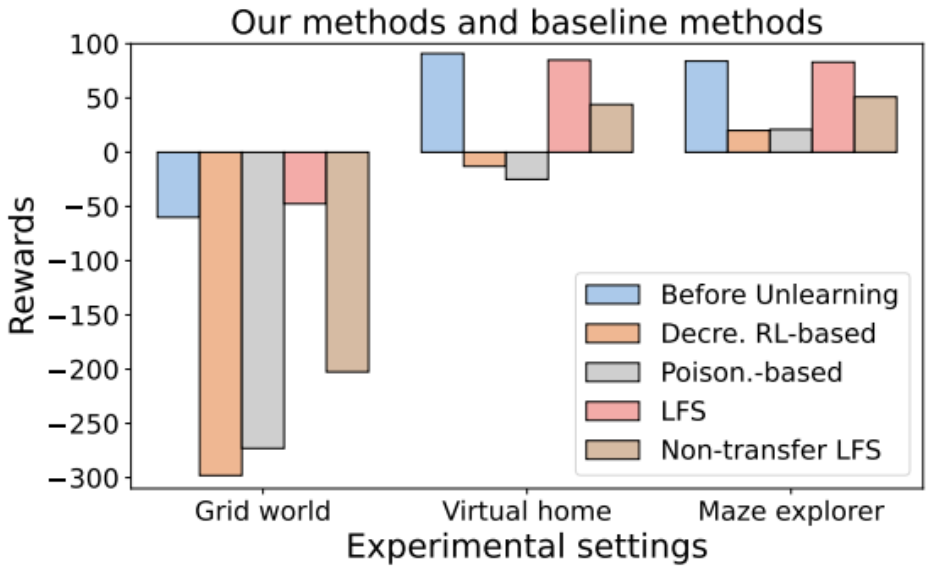
- **Base RL algorithm**
  - DQN for main experiments
  - DDPG, PPO for general performance

Results (I) – Unlearning results in goal-based tasks

- **Proposed methods outperforms the LFS baselines in all tasks**
  - These results indicate that the proposed methods are more effective at forgetting in the unlearning environment.
  - The unlearning results of the non-transfer LFS method underscore the effectiveness of using an inverse loss to minimize the agent's cumulative reward in the unlearning environment, compared to naive LFS.



(a) Average steps of the four methods

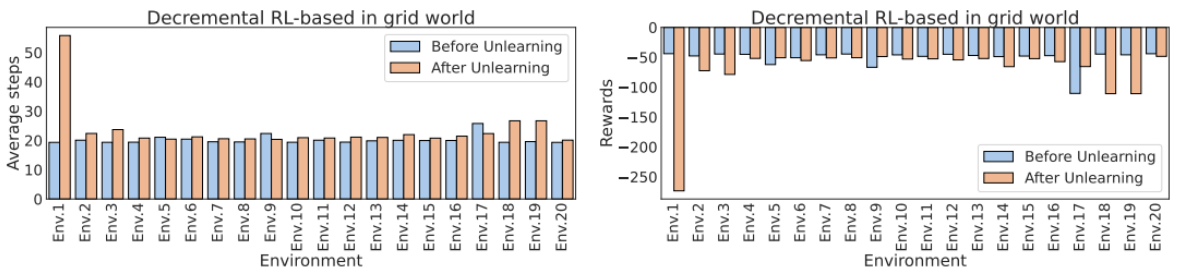


(b) Rewards of the four methods

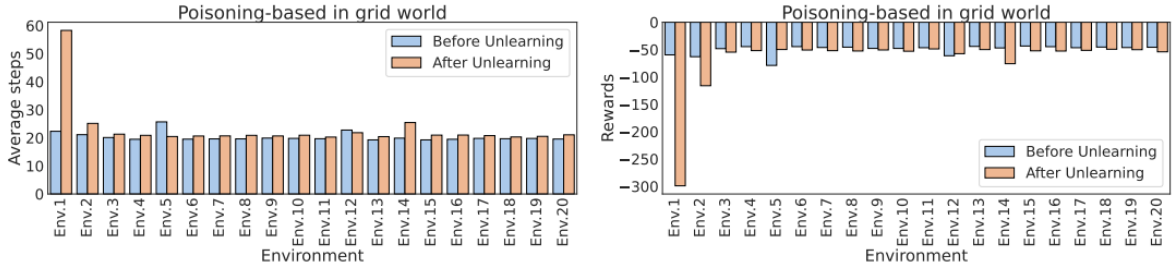
Evaluation results in three goal-based tasks: Grid world, Virtual home, Maze explorer

Results (II) – Unlearning results in grid-world

- **While both methods achieve effective forgetting in the unlearning environment, they differ in their ability to preserve performance in the remaining environments.**
  - The poisoning-based method maintains nearly unchanged performance before and after unlearning, demonstrating greater robustness against the over-unlearning issue
  - Whereas the decremental RL-based method shows **performance degradation** in some remaining environments.



(a) The average number of steps before and after unlearning (b) The average rewards before and after unlearning



(a) The average number of steps before and after unlearning (b) The average rewards before and after unlearning

Unlearning results in Grid World, where env 1 is the unlearning env. Left: Decremental RL method, Right: Poisoning env method

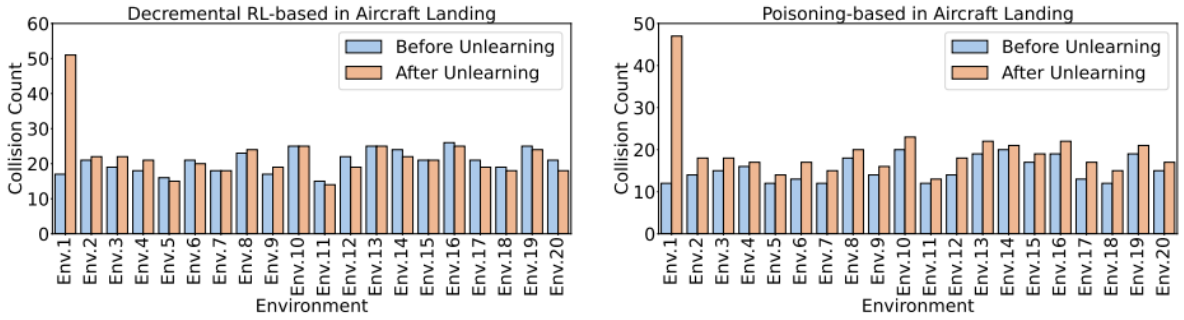
Results (III) – Unlearning results in recommendation and aircraft landing task

- **Results show that recommendation accuracy for the unlearned user drops significantly under the proposed methods.**
  - Meanwhile, rewards fall sharply, reflecting increased randomness in recommendation.
  - Notably, performance for other users remains unaffected, demonstrating the targeted nature of both unlearning methods.

Methods	Performance for the unlearned user		Average performance for the remaining users	
	Accuracy	Reward	Accuracy	Reward
Before Unlearning	92.07%	41.42	91.52%	39.9
Decremental RL	68.63%	20.03	90.89%	38.82
Poisoning	64.41%	18.25	91.43%	37.17

Unlearning results in recommendation task

- **Results show that the number of collisions tripled after unlearning under both methods.**
  - This underscores the need for extreme caution when applying unlearning in safety-critical scenarios such as aircraft landing.
  - The trade-off between privacy and safety must be carefully managed, as reinforcement unlearning can potentially compromise operational safety.



(a) The number of collisions under the decremental RL-based method (b) The number of collisions under the poisoning-based method

Collision count before and after unlearning in Aircraft landing task

Thank you!

Q&A