# Advanced Topic in Reinforcement Learning (048721)

## Submitted by: Eden Grad 212330948

There Is No Turning Back: A Self-Supervised Approach for Reversibility-Aware Reinforcement Learning (NuerIPS 2021) - [link](https://proceedings.neurips.cc/paper/2021/hash/0e98aeeb54acf612b9eb4e48a269814c-Abstract.html)

## Paper summary –

### Introduction

The authors of this article address the issue of estimating the reversibility of actions in the context of Reinforcement Learning. Irreversible outcomes can have positive or negative consequences and decision-makers tend to evaluate the outcomes of irreversible decisions more carefully. The authors propose that incorporating reversibility into decision-making can lead to safer and more efficient behaviors in environments with intrinsic risk factors.

However, estimating the reversibility of actions is challenging and requires planning and causal reasoning in large dimensional spaces. Instead, the authors propose a simpler approach, where they learn the temporal order of events directly from the agent's experience and consider transitions with high temporal confidence to be irreversible.

The contributions of this work include formalizing the link between reversibility and precedence estimation, proposing a practical algorithm to learn temporal order in a self-supervised way through simple binary classification, and proposing two exploration and control strategies that incorporate reversibility. The authors illustrate the practical use of these strategies in synthetic and complex tasks, such as the Sokoban puzzle game.

### Related Work

The text discusses a new approach to reinforcement learning that explicitly models the reversibility of transitions and actions to guide exploration and control. The authors review previous works related to leveraging reversibility in RL, safe exploration, and self-supervision from the arrow of time. The authors note that their approach differs from previous works in its explicit goal of finding irreversible transitions or actions through temporal order prediction, as opposed to learning reachability, generating learning data, or detecting irreversibility through a potential function. The authors also note that their method does not require a preexisting safe policy, restrict policy search to ergodic policies, or assume prior knowledge of environment dynamics.

Some of the previous works mentioned include:

*Kruusmaa et al.* [1] *-* This work is focused on estimating the reversibility of state-action pairs for robotic systems, in order to avoid performing irreversible actions that could damage the robot or its environment. Their approach involves collecting explicit state-action pairs and their reversal actions, which can be a limitation in scaling to larger environments.

Reachability-based exploration: Several works have used reachability as a curiosity bonus for exploration, including Savinov et al. [2], Badia et al. [3, 4]. These approaches use estimated distance to previous states as a measure of novelty, and reward agents for exploring novel states. Reachability and reversibility are related in that irreversible actions can lead to states from which previous states are unreachable. However, these approaches are motivated by exploration and curiosity, rather than explicitly learning reversibility.

Nair et al. [5]: This work is focused on generating realistic trajectories that reach a goal state by learning to reverse trajectories that start from a goal state. Unlike the current work, they use reversibility for generating learning data, rather than directing exploration and control.

Rahaman et al. [6]: This work proposes to learn a potential function of the states that increases with time, which can detect irreversibility to some extent. However, the potential function is learned using trajectories sampled from a random policy, which can be problematic for tasks where a random agent may fail to cover interesting parts of the state space. In comparison, the current work does not use a potential function and learns jointly with the RL agent.

Safe exploration: Safe exploration aims to ensure that RL agents do not perform actions that have negative or unrecoverable effects that would outweigh the long-term value of exploration. Previous works have developed distinct approaches to avoid irreversible behavior, including incremental updates to safe policies, restricting policy search to ergodic policies, active exploration, and computing regions of attraction.

Self-supervision: Self-supervision has become a central component of modern machine learning algorithms. Using temporal consistency as a source of self-supervision is now ubiquitous, be it to learn representations for downstream tasks or to learn to detect temporal inconsistencies. There are several methods that estimate some aspects of the arrow of time as self-supervision, particularly in video processing literature. These tasks include predicting the direction of time flow, verifying the temporal order of frames, predicting which video clip has the wrong temporal order, and reordering shuffled frames or clips. In RL, self-supervision has gained momentum in recent years, with temporal information being featured. Several works have leveraged temporal consistency to learn useful representations, effectively learning to discriminate between temporally close and distant observations.

### Reversibility – definitions:

Degree of Reversibility –

Given a state and an action we define the degree of reversibility as:

With corresponding trajectory.

Given the action a is reversible iff and irreversible iff .  
In deterministic environments all actions are either reversible or irreversible.

Similarly, we define the degree of reversibility of an action according to a fixed policy as:

Precedence Estimation –

Suppose we have two states and and we want to be able to establish precedence, who comes first on average. That is, we would like to estimate   
Accordingly, the precedence estimator which, using a set of trajectories, learns to predict which state of an arbitrary pair is most likely to come first.

To do so, we establish the finite-horizon precedence estimator:

Conceptually, given two states states and , the precedence estimator gives an approximate probability of being visited after , given that both and are observed in a trajectory.

Using the estimator, we define the empirical reversibility:

Conceptually, given that we start in and take the action , the empirical reversibility measures the probability that we go back to , starting from a state that follows

### Paper novelty –

As mentioned in the introduction, the paper has two main novelties. There is theoretical novelty, the new proposed way to quantify reversibility using precedence estimation. Also, there is a more practical novelty, of two suggested algorithms that use reversibility to improve model safety and exploration efficiency.

Theoretical novelty - Estimating Reversibility from Precedence:

The main two propositions proved in the article that links reversibility to precedence:

**Proposition 1.** Given a policy , a state and an action , we have:

That is, the empirical reversibility, calculated using precedence estimation is an upper bound of half the real value of the reversibility.   
This is useful because it provides a way of using the empirical reversibility to detect action which are irreversible or hardly reversible. Since, . Thus, provides a sufficient condition to detect actions with low degrees of reversibility.   
This result gives a way to detect actions that are irreversible given a specific policy followed by the agent. Nevertheless, we are generally interested in knowing if these actions are irreversible for any policy .

The next proposition does exactly that:  
**Proposition 2.**   
Given a state , an action a that is reversible in steps and a policy . If is stochastic enough, that is such that we have . Then .

This proposition gives a practical way of detecting irreversible moves. If for example for some , we can be sure that action a is not reversible in k steps.

Practical novelty - Reversibility-Aware RL algorithms

The first step is to learn to rank events chronologically by binary supervised classification. This is done by sampling pairs of observations uniformly in a sliding window on observed trajectories and training a classifier to predict which observation comes first. The training can be done offline using a previously collected dataset or online during the learning of the RL agent. The classifier is implemented using a network that creates separate embeddings for the pair of observations, which are then concatenated and fed to a separate feed-forward network. The output of this network is passed through a sigmoid to obtain a probability of precedence. The classifier is trained by minimizing the negative log-likelihood against the actual temporal order of the sampled pairs.

However, there is a risk of overfitting to the particularities of the agent's behavior, which can result in the learned reversibility being specific to that agent and not generalizable. To address this risk, the writers are making several assumptions about the agents the method is applied to. First, they assume that agents are learning and have policies that change through interactions in the environment. Second, they assume that agents have an incentive not to be too deterministic. To implement this second assumption, they suggest using an entropic regularization in the chosen RL loss, which is a common design choice in modern RL methods.

Then we can determine whether a transition (and its implicit sequence of actions) is irreversible. This is done by estimating the precedence probability between the starting observation and the resulting observation of the transition using the classifier. If the estimated probability is superior to a chosen threshold, the transition is deemed irreversible. The threshold can be adjusted to cover a range of scenarios, from pure irreversibility to soft irreversibility. Different tasks may require different levels of tolerance for irreversible behavior. For instance, tasks involving human safety might call for absolute zero tolerance for irreversible decision-making, while tasks involving a robot getting stuck might call for some tolerance.

The paper proposes two algorithms, Reversibility-Aware Exploration (RAE) and Reversibility-Aware Control (RAC), both of which are based on reversibility estimation. The basic idea is to estimate the reversibility of a pair of consecutive observations or a set of available actions and use this estimation to create an auxiliary reward function that encourages the agent to take more reversible actions.

RAE estimates the reversibility of consecutive observations and creates an auxiliary reward function based on the estimated reversibility and a fixed threshold. The agent optimizes the sum of the extrinsic and auxiliary rewards. The specific function used by RAE penalizes irreversible transitions but could encourage such transitions if the task requires it.

RAC estimates the reversibility of available actions and "takes control" if the action sampled from the policy is not reversible enough. "Taking control" can involve various strategies, but in practice, the paper uses rejection sampling, where the policy is repeatedly sampled until a reversible action is found.

RAE is more suitable for directed exploration, where irreversible behavior is permitted if the benefits outweigh the costs. In contrast, RAC is more suitable for safety-first, real-world scenarios, where irreversible behavior is to be avoided entirely. RAC can be especially effective when pre-trained on offline trajectories because it is then possible to generate fully reversible, safe behavior from the very first online interaction in the environment.

## My addition: Further Empirical Experiments

The idea in the paper seems too good to be true at first. For every RL environment with unwanted irreversible states, you can add to any of your RL agents a precedence estimator model, and without many changes to the code, it will learn the irreversible state and will help your agent avoid them (either by adjusting the rewards – RAE, or by forcing the agent not to take the irreversible actions RAC). Thus, getting safer, more efficient agents. Furthermore, there are theoretical proves and empirical results showing that the precedence estimator indeed detect the irreversible states.

HOWEVER, there is no mention of the amount of data the precedence estimator needs to be able to learn the irreversible state. In the theoretical part there is no reference to the training and in the empirical part, in all experiments there is no limitations on the amount of learning data.

That raises the question, should we still use the reversibility-aware RL algorithms if we don’t have a large training dataset? Well, one might ask why not? To that I raise the following concern – when we don’t have enough training data the precedence estimator might not learn the irreversible states correctly and without us knowing, direct the agents away from states which are actually reversible that are crucial for the environment.

To test this hypothesis, I will conduct several empirical studies, using a simple environment where I can test if the learned states of the precedence estimator are indeed irreversible – the classic balancing task cartpole. First, I will repeat the paper experiment on the cartpole – running an agent on cartpole without the cartpole standard reward instead, replace it with the intrinsic reward of RAE but with less training timestamps than in the paper. In the second experiment, I will check the model reversibility prediction (of the precedence estimator) of states based on the angle of the pole relatively to the cart position. As for the final experiment, I will compare a simple agent trained on small training data to the same agent trained on the same amount of data with added precedence estimator and the RAE algorithm.

Unfortunately, the paper did not provide code implementation for the methods introduced.

Hence, I got the opportunity to implement all myself.

First, I had to implement a simple learning RL agent so I can test the addition of the reversibility methods on to check improvement. (Either modifying the reward with RAE or limiting the action space with RAC). Then, I had to implement the reversibility methods. Here I decided to focus on RAE because I believe it is the more interesting method, as it allows more freedom and slightly modifying the rewards seems more intuitive.

For the RL agent I've implemented an DQN agent, that consists of 3 linear layers with ReLU activation and embedding dimension of 64 for each hidden layer, discount factor of 0.99. For added stability, I Added a target network to compute the states values while the main agent learns the policy. The target network is updated with a soft update.

Implementing the RAE model was not too difficult. For the precedence estimator, I've implemented a Siamese network for the embedding of the states, consists of 2 linear layers with ReLU activation with embedding dimension of 64, then concatenating the embedding passing them through a linear projection layer to size of 1, and through a sigmoid function to receive the reversibility probability.

I've trained the precedence estimator as described in the article, sampling pairs of state from the same trajectory (from the trajectories the agent collected) then shaffling them. The loss is a cross entropy between the precedence estimator prediction and the real order of the states (before the shaffle).

When using the RAE algorithm, the agent receives the prediction of the precedence estimator, and decrease the reward if the prediction surpass some value .

For my experiments I used .

Results:

The first experiment – cartpole only with the RAE reward:

Chart, histogram

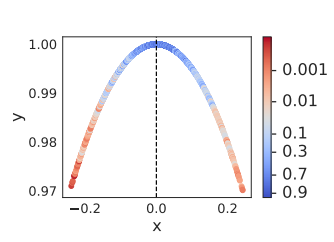
Description automatically generated

(The maximum episode length in cartpole is 500)

We can see that the indeed the at least some of the irreversible states were learned at some point because the episode durations increased with only loss to avoid the learned irreversible states. However, at some point the episode duration length starts decreasing. I couldn’t figure out why. In the next experiment we may be able to shed light on these results.

The second experiment - reversibility prediction based on the angle of the pole.

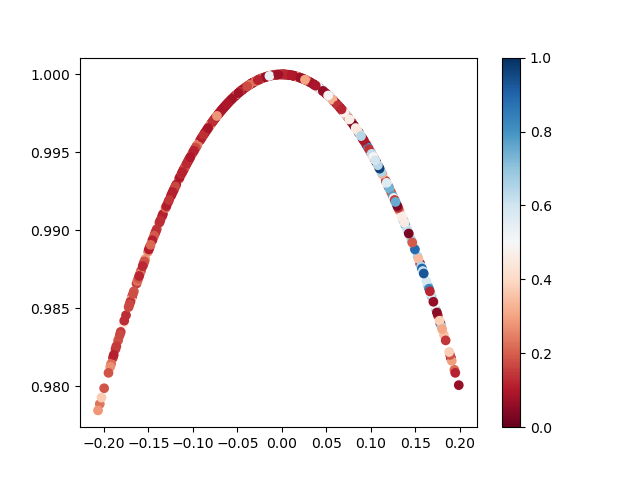
We would hope to see results similar to those of the paper:



We can see that the extreme angles states are irreversible since in those states the pole is close to fall and are likely to be irreversible.

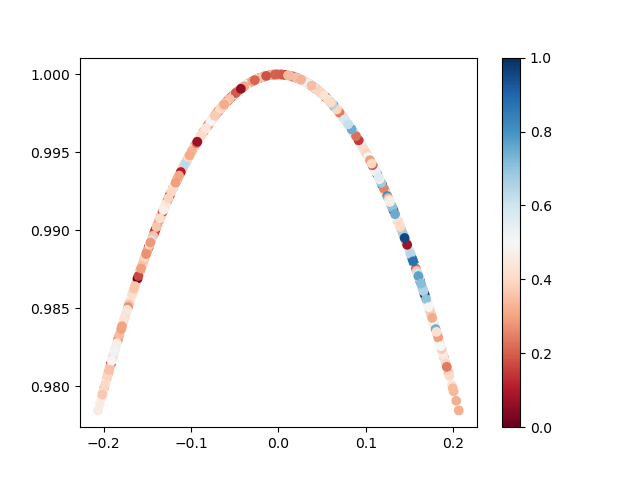
The results of the paper are after 500,000 training steps.

When I run on lower amounts of steps, I got worse results:

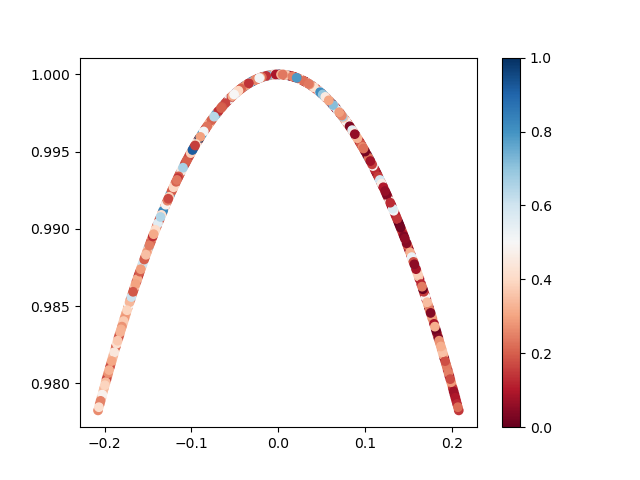


With 100 training steps:

Almost all states are predicted irreversible.

With 500 training steps:

Similar results, However the estimator values are less extreme and its less sure about the reversibility.



Only with 5000 steps we start to see some change:

Some of the angles in the top have a high reversibility score. Still, not close to the high to how it looks with 100,000 training steps.

With these results, I am not sure how it was able to learn in the previous experiment.

Third experiment - comparing a simple agent to the same agent trained with RAE

Here we will be able to tell for sure whether the RAE model helps or not with small data. I run the same agent for 1000 steps. One time with RAE and once without. I averaged the runs of 10 different randomization seed values over the episodes' durations:

As we can see, my suspicioun was justified. Adding the RAE algorithm caused worse results than the simple DQN agent.

Conclusions:

As we see from the paper and the first experiment, the reversibility aware algorithms are viable and can definitely be used to improve the agent, that is only if we have enough data to learn the reversibility of the states.

From the second and third experiments, we can see that if we are not sure if the reversibility of the states is well estimated, it is better to not use it as it may actually hurt the agent instead of improving the efficiency.

Thus unfortunately, in problems of small data, I don’t believe this novel and promising idea should be used.

References:

1. M. Kruusmaa, Y. Gavshin, and A. Eppendahl. Don’t do things you can’t undo: Reversibility models for generating safe behaviours. In International Conference on Robotics and Automation, 2007.
2. N. Savinov, A. Raichuk, R. Marinier, D. Vincent, M. Pollefeys, T. Lillicrap, and S. Gelly. Episodic curiosity through reachability. In International Conference on Learning Representations, 2019.
3. A. P. Badia, P. Sprechmann, A. Vitvitskyi, D. Guo, B. Piot, S. Kapturowski, O. Tieleman, M. Arjovsky, A. Pritzel, A. Bolt, et al. Never give up: Learning directed exploration strategies. In International Conference on Learning Representations, 2020.
4. A. P. Badia, B. Piot, S. Kapturowski, P. Sprechmann, A. Vitvitskyi, Z. D. Guo, and C. Blundell. Agent57: Outperforming the atari human benchmark. In International Conference on Machine Learning, 2020.
5. S. Nair, M. Babaeizadeh, C. Finn, S. Levine, and V. Kumar. Time reversal as self-supervision. In International Conference on Robotics and Automation, 2020.
6. N. Rahaman, S. Wolf, A. Goyal, R. Remme, and Y. Bengio. Learning the arrow of time for problems in reinforcement learning. In International Conference on Learning Representations, 2020.