Proceedings of Machine Learning Research vol 242:68–79, 2024 6th Annual Conference on Learning for Dynamics and Control

An Investigation of Time Reversal Symmetry in Reinforcement Learning

Brett Barkley BBARKLEY@UTEXAS.EDU

Amy Zhang AMY.ZHANG@AUSTIN.UTEXAS.EDU

David Fridovich-Keil DFK@UTEXAS.EDU

The University of Texas at Austin, Austin, TX 78712

Editors: A. Abate, M. Cannon, K. Margellos, A. Papachristodoulou

Abstract

One of the fundamental challenges associated with reinforcement learning (RL) is that collecting sufficient data can be both time-consuming and expensive. In this paper, we formalize a concept of time reversal symmetry in a Markov decision process (MDP), which builds upon the established structure of dynamically reversible Markov chains (DRMCs) and time-reversibility in classical physics. Specifically, we investigate the utility of this concept in reducing the sample complexity of reinforcement learning. We observe that utilizing the structure of time reversal in an MDP allows every environment transition experienced by an agent to be transformed into a feasible reverse-time transition, effectively doubling the number of experiences in the environment. To test the usefulness of this newly synthesized data, we develop a novel approach called time symmetric data augmentation (TSDA) and investigate its application in both proprioceptive and pixel-based state within the realm of off-policy, model-free RL. Empirical evaluations showcase how these synthetic transitions can enhance the sample efficiency of RL agents in time reversible scenarios without friction or contact. We also test this method in more realistic environments where these assumptions are not globally satisfied. We find that TSDA can significantly degrade sample efficiency and policy performance, but can also improve sample efficiency under the right conditions. Ultimately we conclude that time symmetry shows promise in enhancing the sample efficiency of reinforcement learning and provide guidance when the environment and reward structures are of an appropriate form for TSDA to be employed effectively.[1]

Keywords: Reinforcement learning, sample-efficient learning, data augmentation

1. Introduction

In this paper, we investigate time reversal symmetry in a Markov decision process (MDP) and apply it as time symmetric data augmentation (TSDA) in deep reinforcement learning (DRL). DRL is a machine learning paradigm that uses deep learning techniques and an agent's interactions with its environment to learn a policy that maximizes rewards. However, the efficacy of DRL, particularly in the real world, often depends upon the ability to collect sufficient data, which can be challenging due to its high cost and/or limited availability.

Time reversal symmetry is particularly promising because for every rollout of the policy a feasible reverse time trajectory can be generated. This trajectory provides counterfactual, and likely off-policy, experiences that normally would require additional interactions

1. The full version of the paper, including the appendix, can be found at https://arxiv.org/abs/2311.17008

with the environment to obtain. This information can be utilized by downstream learning algorithms in a number of ways; in this paper, we investigate data augmentation since it is relatively algorithm-agnostic. In particular, TSDA enables us to effectively double the amount of experiential data in an off-policy manner.

The central hypothesis of this work is that utilizing time symmetric data can improve sample efficiency in reinforcement learning if the underlying environment is time reversible. To acquire this data, we first provide explicit conditions that allow an observed state transition to be reversed in time through a conjugate transformation. Although these requirements are strict, and are readily violated by physical phenomena such as friction and impulsive contact forces, we find empirically that time symmetric transitions can still remain valuable even in environments that do not strictly satisfy these constraints. This evidence suggests that using TSDA will directly extend to a wide variety of real-world learning problems, in which physical environment dynamics are rarely precisely time symmetric.

To evaluate the utility of the time reversal phenomenon, we conduct extensive experiments on time symmetric and asymmetric reinforcement learning (RL) benchmarks. These investigations show that using time reversibility for data augmentation can improve sample efficiency in both ideal environments and where optimal policies are incentivized to avoid time asymmetric transitions. However, in time asymmetric environments without this reward structure, we find that synthetic transitions hinder learning due to their physical infeasibility. Overall, our investigations of TSDA demonstrate that it can be useful when generating counterfactual transitions, combines the strengths of model-based and model-free techniques, and retains the simplicity and flexibility of model-free algorithms.

2. Time Reversal, Symmetry, and Data Augmentation in Learning

Recent related work has proposed using time reversal for machine learning in both high dimensional (pixel-based) and low dimensional (proprioceptive) state tasks, and the success of these methods motivates our work. In particular, Nair et al. (2020) proposed a self-supervised approach to goal-conditioned RL which exploits reverse-time trajectories. Similarly, Edwards et al. (2018) incorporate a learned backward-dynamics model which aids in the computation of these reverse-time trajectories. However, neither Nair et al. (2020) nor Edwards et al. (2018) formally consider or exploit time symmetric state transitions.

Our work also builds upon that of Grinsztajn et al. (2021), which defines degree-N action reversibility, i.e., conditions under which an action can be "undone" with a sequence of N actions. The dynamically action reversible Markov decision process (DARMDP) introduced in this paper can be seen as an adaptation of this concept. Our work is also comparable to that of Cheng et al. (2023), though their work focuses on offline settings and enforces time symmetry consistency between the forward and reverse latent dynamics generated by a learned model. Recent work also proposed exploiting symmetry as a form of data augmentation (Lin et al., 2020; Mavalankar, 2020); however, these approaches require an expert to exhaustively identify such symmetries. Many recent methods also focus on using symmetry for constrained learning, rather than data augmentation, to improve efficiency in DRL. These ideas tend to focus either on policy and loss symmetry constraints (Abdolhosseini et al., 2019; Yu et al., 2018) or underlying neural network structure (Van der Pol et al., 2020; Yu et al., 2018). Both approaches are promising for situations where an optimal policy exhibits symmetry properties; however, these approaches still require substantial expert knowledge and the addition of specialized structures in the learning formulation.

Finally, data augmentation (DA) has been increasingly used in RL, with examples including image transformations (Laskin et al., 2020; Yarats et al., 2021b; Kostrikov et al., 2021; Yarats et al., 2021a), exploiting symmetry in proprioceptive state (Silver et al., 2016; Lin et al., 2020), and adding noise to proprioceptive states (Laskin et al., 2020). Data augmentation techniques that are independent of symmetry such as image transformations and state noise (Laskin et al., 2020; Kostrikov et al., 2021) require minimal knowledge of the environment, in contrast symmetry exploiting methods, e.g., geometric or reflection symmetry (Lin et al., 2020; Mavalankar, 2020), which require a priori environment knowledge.

3. Background

This section introduces four essential concepts for understanding time reversal symmetry in MDPs: the physical phenomenon of time reversal symmetry, reversible Markov chains (MCs), dynamically reversible MCs, and the structure of an RL problem.

3.1. Time Reversal Symmetry in Dynamical Systems

In classical mechanics, the concept of time symmetry asserts that the evolution of a physical process remains equally feasible even when the direction of time is reversed. This concept is generally associated with conservative systems wherein total energy remains constant, e.g. for systems without frictional or impulsive contact forces. One way to describe this phenomenon is using the Hamiltonian of a dynamical system, H. The Hamiltonian takes as input position and momentum, $\mathbf{q}, \mathbf{p} \in \mathbb{R}^n$, and can be used to generate the equations of motion that govern a system's evolution (Lamb and Roberts, 1998):

$$\frac{\partial H}{\partial \mathbf{p}} = \frac{d\mathbf{q}}{dt}, \frac{\partial H}{\partial \mathbf{q}} = -\frac{d\mathbf{p}}{dt}. \tag{1}$$

If the system modeled by H is conservative, the Hamiltonian will remain constant as a function of time and be an even function with respect to the momentum vector: $H(\mathbf{q}, -\mathbf{p}) = H(\mathbf{q}, \mathbf{p})$. Under these conditions, the physical relations in (1) will remain unchanged under the time reversal transformation:

$$T : (\mathbf{q}, \mathbf{p}, t) \to (\mathbf{q}, -\mathbf{p}, -t). \tag{2}$$

This invariance implies that any forward time trajectory for a conservative system can be reversed in time and momentum to generate an equally feasible trajectory.

3.2. Reversible MCs

A stationary MC is reversible if it satisfies the detailed balance condition (Kelly, 1978):

$$P(s_t)P(s_{t+1}|s_t) = P(s_{t+1})P(s_t|s_{t+1}),\ \forall\ s_t, s_{t+1}, \tag{3}$$

where $P(s_t)$ and $P(s_{t+1})$ are the equilibrium probabilities of states s_t and s_{t+1}, respectively, and $P(s_{t+1}|s_t)$ and $P(s_t|s_{t+1})$ are the probability of transitions from states s_t to s_{t+1} and from s_{t+1} to s_t. For a MC to remain in equilibrium regardless of transition direction, these transition probabilities must be balanced for each pair of states.

3.3. Time Reversal Symmetry in MCs

Dynamically reversible Markov chains (DRMCs) are a discrete-time analog to time reversal symmetry in classical physics (Kelly, 1978). A stationary MC can be transformed into a DRMC by incorporating the concept of conjugate states. Given a state s_t, we obtain its conjugate state using an involution f:

$$s_t^+ = f(s_t). \tag{4}$$

For a MC to be dynamically reversible, conjugate states must satisfy detailed balance:

$$P(s_t)P(s_{t+1}|s_t) = P(s_{t+1}^+)P(s_t^+|s_{t+1}^+),\ \forall\ s_t, s_{t+1}. \tag{5}$$

For physical systems, f corresponds to the continuous-time transformation T introduced in (2), which negates the subset of the state dimensions corresponding to momentum (Lamb and Roberts, 1998; Kerdcharoen et al., 1996) and is generally known a priori. Applying f to a vector of position and momentum results in $f(\mathbf{q}, \mathbf{p}) = (\mathbf{q}, -\mathbf{p})$, which is identical to the time reversal transformation in (2), but without the explicit time negation.

3.4. Reinforcement Learning

RL algorithms aim to learn a policy that maximizes the expected cumulative reward obtained through interactions with an environment (Sutton and Barto, 2018). The agent's interaction can be modeled as an MDP, which is a tuple (S, A, P, R, γ), where S is the state space, A is the action space, $P(s_t|s_{t-1}, a)$ is the probability of transitioning from state $s_{t-1} \in S$ to $s_t \in S$ given action $a \in A$, and $\gamma \in [0, 1]$ is a discount factor that determines the importance of future rewards. In the MDP, each state transition yields a reward $r = R(s)$ and the goal of RL is to find a policy $a \sim \pi(\cdot|s)$ that maximizes the expected cumulative reward: $\mathbb{E}_{s_0 \sim p_0, \pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$, where s_0 is the initial state, p_0 is the initial state distribution, and π is the policy.

4. Dynamically Action Reversible MDPs and Time Symmetric Data

In this section, we present two of our contributions: the concept of a dynamically action reversible Markov decision process (DARMDP) and a corresponding approach for data augmentation in off-policy RL.

4.1. The Dynamically Action Reversible MDP

Consider a physical system such as a spacecraft following a fixed orbit, subject to a gravitational field. The system can be modeled with a dynamically reversible MC due to its conservative nature, but when thrusters are used to change orbit, energy is no longer conserved due to the introduction of a non-conservative force. Nevertheless, the system can still be reversed in time. We can integrate this new notion of time reversibility into an MDP using DARMDPs and discuss the utility of such transitions below.

Definition 1 (DARMDP) *Let A and S be the action and state space of an MDP, $a^+ \in A$ and $s^+ \in S$ be conjugate transformations of the action a and state s, and f an involution on A. An MDP is a DARMDP if $\forall\ s_t, s_{t+1} \in S, \exists\ f(a) = a^+$ s.t. $P(s_t|s_{t-1}, a) = P(s_{t-1}^+|s_t^+, a^+)$.*

To illustrate Definition 1, consider OpenAI's Lunar Lander (Brockman et al., 2016) with $a \in A$ comprising control over the main booster and $s \in S$ representing the altitude of the craft. As the lander descends, the state-action transition is represented by the tuple (s_{t-1}, a, s_t) and the time reversed trajectory is (s_t^+, a^+, s_{t-1}^+). The first tuple represents the lander decelerating during descent under action a, whereas the second tuple represents the time reversed case where the lander instead accelerates during ascent under a^+. If the state space is constrained to cases where no contact is made between the lander and the moon, this MDP is a DARMDP. However, if the state space includes situations where high impulse contact is made with the lunar surface there will be cases where the detailed balance condition in Definition 1 does not hold and the MDP is not a DARMDP.

As shown in the example of the lunar lander, for a DARMDPs, a state-action transition is transformed elementwise into its conjugate counterpart using a state-action involution analogous to that introduced in (4): $(s_t^+, a^+, s_{t-1}^+) = f(s_{t-1}, a, s_t)$.

4.2. Using Conjugate Transitions for Data Augmentation

In this section we explore the use of time reversal symmetry in DARMDPs as a form of data augmentation to aid in training an RL agent. While investigating time symmetric data augmentation (TSDA) we make the following assumptions:

Assumption 1 *The action is unchanged under time reversal, i.e.* $a = a^+$*. This ensures that the policy only needs to be queried once for a conjugate pair of transitions, but is otherwise only included for simplicity of exposition.*

Assumption 2 *We have a priori knowledge of the MDP reward function. Consequently, the conjugate transition's reward can be assessed directly.*

TSDA uses the time symmetry involution introduced in Section 4.1 to produce reverse time transitions for each policy rollout in the DARMDP. These reverse time rollouts are counterfactual, in the sense that they are generally off-policy due to differences between $\pi(a|s_{t-1})$ and $\pi(a|s_t^+)$. The pair of forward and reverse time states, actions, and rewards are then added to the agent's replay buffer and subsequently used to learn a policy via an off-the-shelf RL algorithm.

TSDA is especially useful because the conjugate state-action involution f provides conjugate transitions for proprioceptive and pixel-based states with minor modification. Consider first a proprioceptive state transition (s_{t-1}, a_{t-1}, s_t) with reward r_{t-1}. A feasible conjugate transition is generated by simply applying the state-action involution f from Section 4.1, with the action a held constant per Assumption 2, and reward given by $r_{t-1}^+ = R(s_{t-1}^+)$. In the case of pixel-based states, the transformation reverses the order of a sequence of pixel-based states without further modification, analogous to playing a video backward.

5. Experimental Setup

To evaluate the role of time symmetry in data augmentation for RL we conduct a series of numerical experiments. In particular, in Section 6 we investigate the following hypotheses:

Hypothesis 1 *Utilizing time reversal symmetry for data augmentation can improve sample efficiency in reinforcement learning if the underlying environment is time reversible.*

Hypothesis 2 *TSDA can lead agents to favor early exploitation over exploration in time symmetric environments, e.g., when important exploratory actions lie at the edge of the action space.*

Hypothesis 3 *TSDA enhances sample efficiency in time asymmetric MDPs if the reward structure incentivizes policies to avoid time asymmetric transitions, e.g., when high-impulse contact is a feature of a poorly performing policy.*

Hypothesis 4
TSDA becomes ineffective when time reversal symmetry is frequently and significantly broken, e.g. in scenarios requiring high-impulse contact for an optimal policy.

5.1. Environment Setup

We use several simulation environments from the DeepMind Control Suite (Tassa et al., 2018) and OpenAI Gym (Brockman et al., 2016) to benchmark TSDA. DeepMind's Cartpole swing-up is used for testing Hypotheses 1 and 3 by varying the friction coefficient, whereas Lunar Lander and Walker walk/run are used to test Hypotheses 2 and 4, respectively. In order to increase the sample complexity of time symmetric benchmark tasks, we designed a customized environment based upon the work in Tassa et al. (2018). This environment consists of a fully actuated n-link manipulator where each joint actuator has a configurable maximum torque. To study the role of actuator authority in Hypothesis 2, we consider three levels of torque limits in this environment: underpowered (`uM`), intermediate (`iM`), and overpowered (`oM`). The task is to extend the manipulator arm fully upright, with a reward structure identical to the `cart-k-pole` example introduced in Tassa et al. (2018).

Table 1 enumerates these environments and their time symmetric properties. We also include the episodic return at which the task is considered to be solved. A larger discussion about rewards and solving tasks is included in the appendix of the full version of this work.

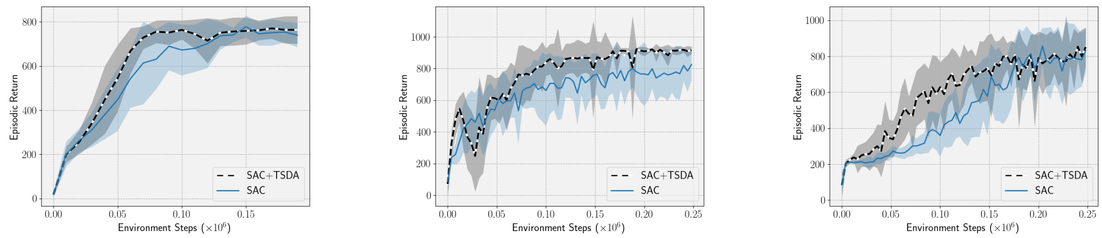
Environment	Friction	Contact	Episodic Return for Solution
DeepMind Cartpole	✓	✗	750
Lunar Lander	✓	✓	200
DeepMind Walker Stand/Run	✓	✓	800/600
Custom N-link Manipulator	✗	✗	800

Table 1: Summary of benchmark environments. Those with two checkmarks are time symmetric.

5.2. Algorithmic Baselines and Evaluation Metrics

In our empirical studies, we employ the soft actor-critic (SAC) algorithm (Haarnoja et al., 2018), with and without TSDA. When state is encoded as a sequence of images we use SAC+AE (Yarats et al., 2021b), which augments SAC with a regularized convolutional neural network (CNN) based encoder/decoder. For proprioceptive state, the autoencoder is removed and states are passed directly to the SAC algorithm as in Haarnoja et al. (2018).

During training, the policy is evaluated every 4000 environment state transitions and the average return over 10 episodes is computed. Training is repeated for 10 random seeds with the mean and standard deviation for episodic return reported. Relative performance between SAC and TSDA+SAC is determined by comparing both the averaged episodic



(a) (b) (c)

Figure 1: MDPs where TSDA data augmentation definitively improves sample efficiency. Results are averaged over 10 seeds with shading indicating one standard deviation away from the mean. a) Cartpole swing-up from pixels. b) 3-link `oM` environment. c) 4-link `iM`.

return at training convergence and the number of environment steps required to pass an episodic return solution threshold as shown in Table 1.

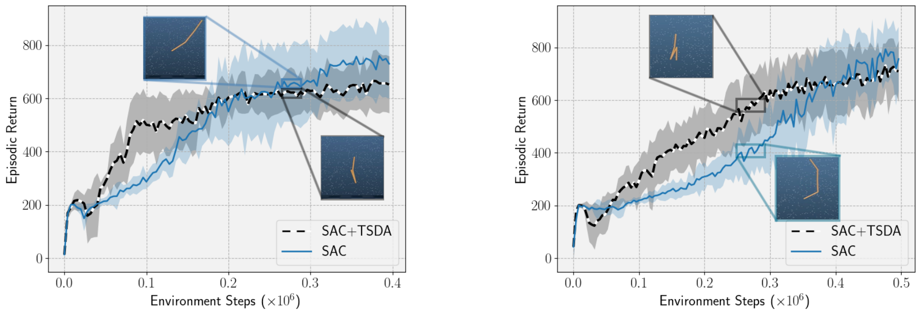
6. Results and Analysis

6.1. Improved Performance in Time Symmetric Environments

We begin by showing three cases that support Hypothesis 1. In these cases, TSDA is applied in a time symmetric environment and improves the sample efficiency of learning, as summarized in Figure 1. Figure 1(a) shows learning progress for a frictionless cartpole swing-up using pixel-based state and the CNN encoder/decoder structure of Yarats et al. (2021b). By leveraging both reverse time-transitions in pixels and true environment steps, the agent is able to find a policy that solves the task with 50% fewer environment steps.

In Figure 1(b), we observe that in the early stages of training with TSDA, there is a noticeable increase in the variance of the evaluated policies for the 3-link overpowered manipulator environment (`oM`). However, as training progresses, TSDA+SAC effectively solves the task with ∼62% fewer samples and reaches an averaged episodic return that is ∼12% higher than SAC. TSDA performs well in both cases because the environment is time symmetric and the agent has access to a large amount of actuator authority, which leads to increased diversity in the transitions augmented by TSDA. A lack of diverse transitions becomes particularly important for agents acting in environments with less actuator authority, which we shall study shortly. However, for Figures 1(a) and 1(b), the introduction of TSDA allows the agent to explore more without experiencing any additional physical transitions, and the agent subsequently exploits this additional information in policy learning.

As mentioned in Hypothesis 2, when actuator authority is reduced, exploratory actions used in `oM` may become infeasible, which leads to degraded performance in TSDA. This effect is illustrated in Figure 1(c) for an `iM` with 4-links where the actuator authority reduction is accompanied by a reduction in the performance gap between SAC and TSDA+SAC. Specifically, the slope of the TSDA+SAC learning curve is significantly larger than SAC early in training, but both require a similar number of samples. This effect implies that the time reversed transitions in this smaller action space are nearer to on-policy and less diverse,



(a) (b)

Figure 2: TSDA data augmentation leads to sub-optimal policies in training under limited actuator authority. a) 3-link `uM` environment. b) 4-link `uM` environment.

which incentivizes exploitation over exploration. We provide further evidence in Section 6.2 where we consider a robotic system that has even further degradation in actuator authority.

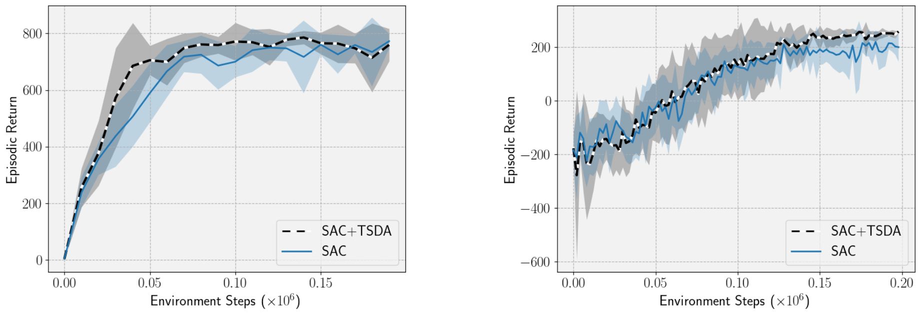
6.2. Time Symmetric Environments with Limited Actuator Authority

In this section we show two cases where an environment is time symmetric, yet TSDA does not improve sample efficiency in solving tasks. Figures 2(a) and 2(b) show cases of the n-link underpowered manipulator where the requirements of time symmetry are satisfied, but TSDA is harmful in achieving optimal policies. In both cases, SAC+TSDA achieves superior episodic returns before convergence, but SAC achieves higher final policy performance.

These results suggest a deeper interaction between actuator authority and TSDA, as mentioned in Hypothesis 2. Here the exploration capabilities of the agent across a single environment transition are degraded since it is operating in the `uM` environment. Consequently, the time-reversed transitions are even closer to on-policy than shown in Figure 1(c), further biasing the agent toward exploitation. This intuition is corroborated by both individual evaluation trajectories available during training and the recorded video of policy evaluation episodes. Snapshots of common states with and without TSDA are overlaid in Figure 2 to illustrate this trend. We note that in the case of TSDA, these states correspond to sub-optimal outcomes that would likely require substantial exploration to escape. Ultimately, in both experiments we find that TSDA helps the agent identify these local optima faster and more frequently, but that their discovery hinders final policy performance.

6.3. Improved Performance in Time Asymmetric Environments

Figure 3 shows two examples that support Hypothesis 3 wherein the uncontrolled MDP is not strictly dynamically reversible, yet augmenting SAC with TSDA still improves performance. The first case, shown in Figure 3(a), is cartpole swing-up using pixel-based state; unlike the results of Figure 1(a) we now drastically increase the coefficient of friction to 2000× its nominal value in (Tassa et al., 2018) and render the environment time asymmetric everywhere in the state space. In such a case we would not expect TSDA to improve sample efficiency since transitions are universally time asymmetric, but instead we see that TSDA+SAC+AE still requires ∼37% fewer samples to solve the task.



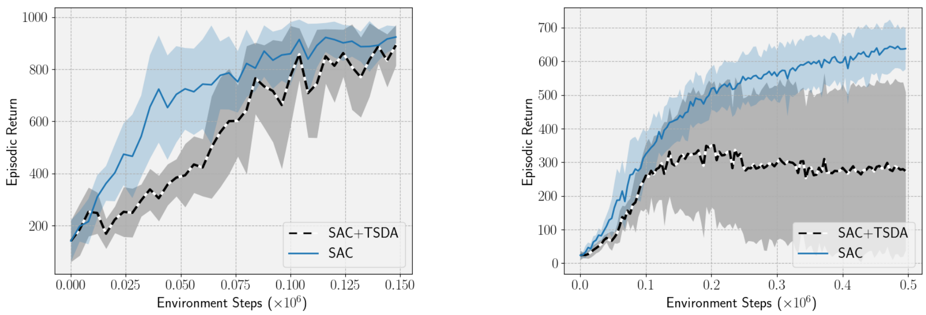
(a) (b)

Figure 3: TSDA improves performance in some (not all) asymmetric environments. a) Cartpole swing-up from pixels with high friction (2000× Mujoco defaults). b) OpenAI Gym Lunar Lander.

One explanation for this phenomenon is that the CNN used to process image-based state in SAC+AE is unable to estimate the velocities in proprioceptive state accurately (cf. (Chua et al., 2018)), and it is these velocities which are most critical for encoding time reversal symmetry, per Section 3.1. As a result, the frictional losses that make the system time asymmetric become difficult to detect when using images as state, making such environments good candidates for TSDA. Figure 3(b) depicts policy training in the OpenAI Gym lunar lander environment (Brockman et al., 2016). TSDA+SAC exhibits a higher sample efficiency relative to SAC, requiring approximately 27% fewer samples to solve the task and achieving a higher averaged episodic return. Initially, this outcome is surprising since the task requires making inelastic contact with the ground in the landing zone, which breaks time symmetry. However, this environment has a reward structure that incentivizes the policy to prioritize transitions that are time reversible. Specifically, the lander is considered to have crashed if its body makes contact with the ground and this results in a very negative reward. Consequently, optimal policies must ensure that the landing struts touch the ground first. Furthermore, the landing struts are relatively fragile if subjected to excessive velocity/impulsivity. This inherent fragility creates an environment conducive to the benefits of TSDA since the optimal policy minimizes energy loss upon ground contact and thus avoids policies that will lead to violations of time symmetry.

6.4. Poor Performance in Time Asymmetric Environments

Finally, we investigate Hypothesis 4, focusing on cases where TSDA leads to reduced sample efficiency and final policy performance due to time asymmetry in the environment. We then provide physical intuition to distinguish these cases from those of Section 6.3 in which TSDA still improves policy performance. Figure 4 shows a comparison of policy training in the Walker `stand` and `run` tasks from Tassa et al. (2018). In Figure 4(a), SAC+TSDA requires 2× more environment interactions to achieve similar levels of performance in averaged episodic return, but eventually solves the standing task. However, SAC+TSDA does not solve the `run` task despite early progress. Both of these tasks necessitate making high-impulse contact to attain high reward. In `stand`, falling is a frequent occurrence and invalidates the time reversed transitions from TSDA due to high impulse collisions.



(a) (b)

Figure 4: TSDA definitively worsens performance in (a) Walker `stand` (b) Walker `run`.

However, as an agent becomes capable fewer falls will occur and transitions will obey time reversal symmetry more frequently, explaining the eventual convergence of SAC+TSDA in Figure 4(a). Similarly, the `run` task in Figure 4(b) augments `stand`'s reward structure to encourage forward velocity. Policy performance in `run` requires competency in `stand`, but also frequent and high impulse collisions between the robot legs and the ground. As a result, the reward structure incentivizes operating in the time asymmetric portions of the MDP, TSDA introduces many infeasible transitions, and erratic policy training follows.

7. Limitations and Conclusions

In this work we formalize a notion of time reversal symmetry for MDPs, and investigate when and how it can be exploited to improve the sample efficiency of reinforcement learning. We show that, when used for data augmentation in time symmetric environments, training reliably converges more quickly and to better policies. However, many real-world environments are not time symmetric due to non-conservative forces, such as contact and friction, and the proposed method for TSDA can diminish performance in such cases (as illustrated in Section 6.4). Interestingly, we identify certain special cases which break the conditions of time symmetry, and in which TSDA still improves performance (cf. Section 6.3). These examples provide evidence for the utility of this phenomenon in real-world robotics; however, an important limitation of TSDA is that we have yet to establish either a fundamental theoretical or real-world experimental understanding of these special cases. Future work should focus on establishing both theoretical and experimental results. Finally, generating time reversed transitions requires knowledge of the reward function during training; an assumption which is not strictly required in many RL algorithms (Sutton and Barto, 2018).

References

Farzad Abdolhosseini, Hung Yu Ling, Zhaoming Xie, Xue Bin Peng, and Michiel Van de Panne. On learning symmetric locomotion. In *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*, pages 1–10, 2019.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

Peng Cheng, Xianyuan Zhan, Zhihao Wu, Wenjia Zhang, Shoucheng Song, Han Wang, Youfang Lin, and Li Jiang. Look beneath the surface: Exploiting fundamental symmetry for sample-efficient offline rl, 2023.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models, 2018.

Ashley D. Edwards, Laura Downs, and James C. Davidson. Forward-backward reinforcement learning, 2018.

Nathan Grinsztajn, Johan Ferret, Olivier Pietquin, Matthieu Geist, et al. There is no turning back: A self-supervised approach for reversibility-aware reinforcement learning. *Advances in Neural Information Processing Systems*, 34:1898–1911, 2021.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

Frank Kelly. Reversibility and Stochastic Networks, 1978. URL http://www.statslab.cam.ac.uk/~frank/BOOKS/book/whole.pdf.

Teerakiat Kerdcharoen, Klaus R. Liedl, and Bernd M. Rode. Bidirectional molecular dynamics: Interpretation in terms of a modern formulation of classical mechanics. *Journal of Computational Chemistry*, 17(13):1564–1570, 1996. doi: https://doi.org/10.1002/(SICI)1096-987X(199610)17:13⟨1564::AID-JCC8⟩3.0.CO;2-Q. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291096-987X%28199610%2917%3A13%3C1564%3A%3AAID-JCC8%3E3.0.CO%3B2-Q.

Ilya Kostrikov, Denis Yarats, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels, 2021.

Jeroen SW Lamb and John AG Roberts. Time-reversal symmetry in dynamical systems: a survey. *Physica D: Nonlinear Phenomena*, 112(1-2):1–39, 1998.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in neural information processing systems*, 33:19884–19895, 2020.

Yijiong Lin, Jiancong Huang, Matthieu Zimmer, Yisheng Guan, Juan Rojas, and Paul Weng. Invariant transform experience replay: Data augmentation for deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4):6615–6622, oct 2020. doi: 10.1109/lra.2020.3013937.

Aditi Mavalankar. Goal-conditioned batch reinforcement learning for rotation invariant locomotion. *arXiv preprint arXiv:2004.08356*, 2020.

Suraj Nair, Mohammad Babaeizadeh, Chelsea Finn, Sergey Levine, and Vikash Kumar. TRASS: time reversal as self-supervision. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 115–121. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9196862. URL https://doi.org/10.1109/ICRA40945.2020.9196862.

David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016. URL http://www.nature.com/nature/journal/v529/n7587/full/nature16961.html.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite, 2018.

Elise Van der Pol, Daniel Worrall, Herke van Hoof, Frans Oliehoek, and Max Welling. Mdp homomorphic networks: Group symmetries in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:4199–4210, 2020.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning, 2021a.

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021b.

Wenhao Yu, Greg Turk, and C Karen Liu. Learning symmetric and low-energy locomotion. *ACM Transactions on Graphics (TOG)*, 37(4):1–12, 2018.