The Primacy Bias in Deep Reinforcement Learning

Evgenii Nikishin[*]
Mila, Université de Montréal
evgenii.nikishin@mila.quebec

Max Schwarzer[*]
Mila, Université de Montréal
max.schwarzer@mila.quebec

Pierluca D'Oro[*]
Mila, Université de Montréal
pierluca.doro@mila.quebec

Pierre-Luc Bacon
Mila, Université de Montréal
pierre-luc.bacon@mila.quebec

Aaron Courville
Mila, Université de Montréal
aaron.courville@umontreal.ca

Abstract

This work identifies a common flaw of deep reinforcement learning (RL) algorithms: a tendency to rely on early interactions and ignore useful evidence encountered later. Because of training on progressively growing datasets, deep RL agents incur a risk of overfitting to earlier experiences, negatively affecting the rest of the learning process. Inspired by cognitive science, we refer to this effect as *the primacy bias*. Through a series of experiments, we dissect the algorithmic aspects of deep RL that exacerbate this bias. We then propose a simple yet generally-applicable mechanism that tackles the primacy bias by periodically resetting a part of the agent. We apply this mechanism to algorithms in both discrete (Atari 100k) and continuous action (DeepMind Control Suite) domains, consistently improving their performance.

Keywords: deep reinforcement learning, overfitting, forgetting

Acknowledgements

The authors thank Rishabh Agarwal, David Yu-Tung Hui, Ilya Kostrikov, Ankit Vani, Zhixuan Lin, Tristan Deleu, Wes Chung, Hattie Zhou, Mandana Samiei, Marc G. Bellemare for insightful discussions and useful suggestions on the early draft and Compute Canada for providing computational resources. This work was partially supported by Facebook CIFAR AI Chair, Samsung, Hitachi, IVADO, and Gruppo Ermenegildo Zegna.

[*] Equal contribution, correspondence to Evgenii Nikishin.

1 Introduction

The primacy bias is a well-studied cognitive bias in human learning [13]. Facing a sequence of experiences, humans often form generalizations based on early evidence which might negatively impact future decision making [18].

The central finding of our work is that deep reinforcement learning (RL) algorithms are susceptible to a similar bias. The primacy bias in deep RL is a tendency to overfit early interactions with the environment preventing the agent from improving its behavior on subsequent experiences. Through a series of controlled experiments, we first expose plausible causes of this phenomenon and show that common algorithmic features such as a high replay ratio [7, 4] and long n-step targets [20] amplify the primacy bias. As a remedy, we then propose a simple *resetting* mechanism, compatible with any deep RL algorithm equipped with a replay buffer, which allows the agent to forget a part of its knowledge.

Despite its simplicity, this resetting strategy consistently improves the performance of agents on benchmarks including the discrete-action ALE [2] and the continuous-action DeepMind Control Suite [21]. Resets impose no additional computational costs and require only two implementation choices: which neural network layers to reset and how often.

2 Preliminaries

We adopt the standard formulation of reinforcement learning [20] under the Markov decision process (MDP) and consider deep RL algorithms where the action-value function $Q_\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(s_t,a_t) | s_0 = s, a_0 = a\right]$ and π (when needed) are modelled by neural network function approximators. We focus on off-policy methods that learn $Q_\pi(s,a)$ though *temporal-difference* (TD) learning [19] and reuse past experiences with a *replay buffer* [14]. The frequency of resampling experience from the buffer is controlled by the *replay ratio* [7, 4] which plays a critical role in the algorithm's performance: higher replay ratios may allow better sample efficiency but incur a risk of overfitting. TD learning can be generalized by using n*-step* targets $\mathbb{E}_\pi\left[r(s_t,a_t) + \gamma r(s_{t+1},a_{t+1}) + \cdots + \gamma^n Q_\pi(s_{t+n},a_{t+n})\right]$ for predicting $Q_\pi(s,a)$. Here, n controls a trade-off between the (statistical) bias of Q_π estimates and the variance of the sum of future rewards.

3 The Primacy Bias

The main goal of this work is to understand how the learning process of deep reinforcement learning agents can be disproportionately impacted by initial phases of training due to an effect called the primacy bias.

The Primacy Bias in Deep RL: *a tendency to overfit initial experiences that damages the rest of the learning process.*

This definition is wide-ranging: the primacy bias has multiple roots and leads to multiple negative effects on the training of an RL agent, but they are all connected to improper learning from early experiences.

The rest of this section presents two experiments intending to demonstrate the existence and behavior of the phenomenon in isolation. First, we show that excessive training of an agent on early interactions can fatally damage the rest of the learning process. Second, we show that data collected by an agent impacted by the primacy bias is adequate for learning, although the agent cannot leverage due to its accumulated overfitting.

3.1 Heavy Priming Causes Unrecoverable Overfitting

One of the crucial algorithmic aspects impacting the primacy bias is degree of the reliance of an agent on early data. It is vital for sample efficiency to leverage initial experiences well, and to this end, the agent may sample from its buffer and update its neural network several times before interacting further with an environment. We hypothesize that such a practice may have severe consequences and probe it to its extreme: could *overfitting on a single batch of early data be enough to entirely disrupt an agent's learning process*?

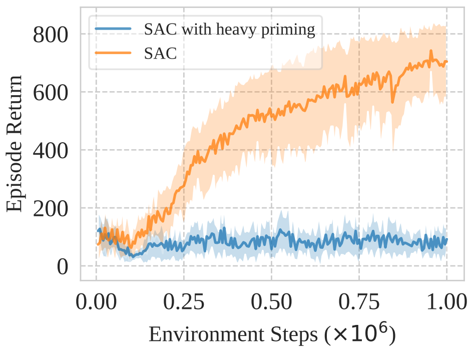

Figure 1: Undiscounted returns on `quadruped-run` for SAC with and without *heavy priming* on the first 100 transitions. An agent extremely affected by the primacy bias is unable to learn even after collecting thousands of new transitions. Mean and std are over 10 runs.

To investigate this question, we train Soft Actor-Critic [6] on the `quadruped-run` environment from DeepMind Control suite (DMC) [21]. We use default hyperparameters, which imply a single update for policy and value functions per step in the environment. Then, we train an identical agent in an experimental condition that we refer to as *heavy priming*: after collecting 100 data points, we update the agent 10^5 times using its replay buffer, before resuming standard training. Figure 1 shows that even after training on almost one million new transitions, the agent with heavy priming is unable to solve the task.

This experiment conveys a simple message: overfitting to early experiences might inexorably damage the rest of the learning process. Indeed, we will see in Section 4 that even a relatively small number of updates per step can cause similar issues. The finding suggests that the primacy bias has compounding effects: an overfitted agent gathers worse data that itself leads to less efficient learning that further damages the ability to learn and so on.

3.2 Experiences of Primed Agents are Sufficient

Once the agent is heavily impacted by the primacy bias, it might struggle to reach satisfying performance. But is the data collected by an overfitted agent unusable for learning? We train a SAC agent with 9 updates per step in the MDP; due to the primacy bias, this agent performs poorly. Then, we initialize the same agent from scratch but use the data collected by the previous SAC agent as its initial replay buffer. Figure 2 demonstrates that returns collected by this agent improve rapidly approaching the optimal task performance.

This experiment articulates that the primacy bias is not a failure to collect proper data per se, but rather a failure to learn from it. The data stored in the replay buffer is in principle enough to have better performance but the overfitted agent lacks the ability to distill it into a better policy. In contrast, the randomly initialized neural networks are not affected by the primacy bias and thus capable of fully leveraging the collected experience. This intuition forms the basis of the algorithmic solution to the issues highlighted above.

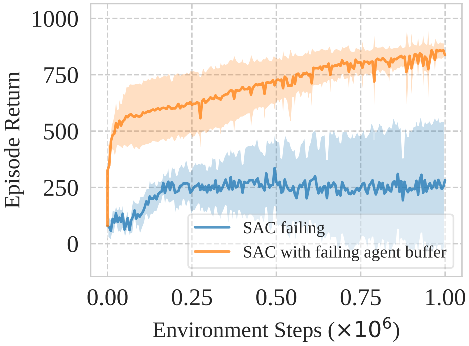

Figure 2: Undiscounted returns on `quadruped-run` for SAC trained with 9 updates per step. `SAC failing` is a standard SAC agent; `SAC with failing agent buffer` is a SAC agent initialized with the replay buffer of the first agent, which allows it to learn quickly. Mean and std are over 10 runs.

3.3 Have You Tried Resetting It?

We now present a simple technique that mitigates the primacy bias. The solution, which we dub *resetting* in the rest of the manuscript, is given by the following recipe:

Addressing the Primacy Bias: *periodically re-initialize the last layers of the agent's neural networks, preserving the replay buffer.*

The next section analyzes both quantitatively and qualitatively the performance improvements provided by resetting in addressing overfitting to early data.

4 Experiments

The goals of experiments are mostly twofold. First, we investigate across different algorithms and domains the effect on performance of using resets as a remedy for the primacy bias; next, we analyze the learning dynamics induced by resetting, including its interaction with critical design choices such as the replay ratio and n-step TD targets.

We focus our experimentation in two settings: discrete control, represented by the 26-task Atari 100k benchmark [9], and continuous control, represented by the DeepMind Control Suite [21]. We apply resets to three baseline algorithms: SPR [17] for Atari, and SAC [6] and DrQ [10] for continuous control from dense states and raw pixels respectively. For SPR, we reset the final layer of a 5-layer Q-network, using three resets spaced 2×10^4 steps apart; for SAC, we reset *the entire policy and value networks* every 2×10^5 steps; for DrQ, we reset last 3 layers of the policy and value networks every 4×10^5 steps. In every case, we also reset target networks. After each reset, we return directly to standard training.

To provide rigorous evaluations of all algorithms, we follow recommendations from [1] and use interquartile mean (IQM) for measuring performance.

Method	IQM
SPR + resets	**0.478** (0.46, 0.51)
SPR	0.380 (0.36, 0.39)
DrQ(ϵ)	0.280 (0.27, 0.29)
DER	0.183 (0.18, 0.19)
CURL	0.113 (0.11, 0.12)
SAC + resets	**656** (549, 753)
SAC	501 (388, 609)
DrQ + resets	**757** (698, 810)
DrQ	570 (473, 665)

Table 1: Point estimates and 95% bootstrap confidence intervals for the performance of SPR, SAC, and DrQ with resets. Results for SPR are computed over 20 seeds per task, and for SAC and DrQ are computed over 10 seeds. Other baselines are taken from [1] and use 100 seeds.

4.1 Resets Consistently Improve Performance

The empirical evidence in Table 1 suggests that resets mitigate the primacy bias and provide significant benefits across environments for the final performance of the agent. Remarkably, the magnitude of improvement provided by resets for SPR is comparable to improvements of prior advances while not requiring additional computation costs.

4.2 The Learning Dynamics of Resetting Agents

At the first glance, resetting may appear as a drastic (if not wasteful) measure as the agent must learn the parameters of the randomly initialized layers from scratch every time. Figure 3 shows a representative example of the learning trajectories induced by resets. Surprisingly, the agent quickly reaches or surpasses its prior performance after each reset. After resetting, the agent is free from the negative priming provided by its past training iterations: it can better leverage the data collected so far, thus improving its performance and unlocking the possibility to generate higher quality data for its future updates. The crucial element behind the success of resetting resides in preserving the replay buffer across iterations allowing the agent to recover.

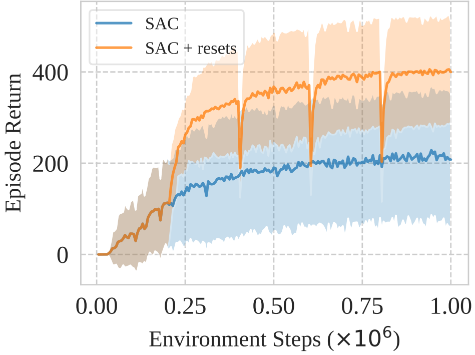

Figure 3: An example showing effects of resets for SAC (32 updates per step, resetting every 2×10^5 steps) on `hopper-hop`. After each reset, performance recovers quickly due to the replay buffer and the resetted agent achieves higher overall returns. Mean and std are over 10 runs.

4.3 The Elements Behind the Success of Resets

We now provide an ablation study aiming at the question: under which conditions resets are maximally impactful?

Replay Ratio. Our initial experiments in Section 3 suggest that the degree of reliance on early data is a critical determinant of the strength of the primacy bias. We vary the replay ratio, the number of gradient steps per each environment step, in SPR and SAC. Figure 4 reports results for SAC while the conclusions for SPR are the same. With fewer updates, resets provide little or no benefit, implying that agents might be underfitting early data. With resets, however, SAC achieves its highest performance at the high replay ratio of 32, where resets increase performance by over 100%. Resets thus allow improving sample efficiency by performing more updates per each data point.

***n*-step targets.** The effect of resets depends on the variance of TD targets. We observe that with high n the agent is more likely to overfit and hence the resetting effect increases. Figure 4 demonstrates the effect size for SPR; for SAC results are similar.

The results with varying replay ratios and n-step targets suggest that resets reshape the hyperparameter landscape creating a new optimum with higher performance.

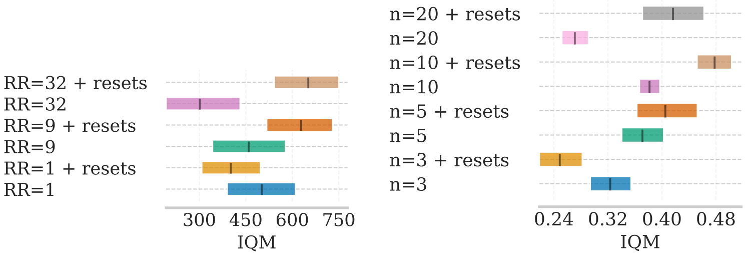

Figure 4: The impact of resets on SAC at different replay ratios (left) and SPR at different n-step return lengths (right). High replay ratio makes the agent overly reliant on the early experince and thus the resets have the largest influence. The effect size of resets is largest for high n since it increases the variance of targets exposing the agent to a risk of overfitting.

What to reset. The number of layers to reset is a domain-dependent choice. We observed the best performance when resetting only the last layer in SPR, while for DrQ resetting the last 3 (out of 7) layers was better. We conduct two additional ablations: resetting or preserving the optimizer state and replay buffer. We find that resetting the optimizer state has essentially no impact because moment estimates are update rapidly. Resetting the replay buffer, however, made it impossible for agents to quickly recover their prior performance.

TD failure modes. Temporal-difference learning can be prone to divergence and collapse to a trivial solution. Once in a failure mode, standard RL optimization struggles to recover from it. Adding resets solves this problem by giving the agent the second chance. We observe that on sparse reward tasks where Q_π might collapse, the buffer contains trajectories reaching the goal state suggesting that *the primacy bias is more an issue of optimization rather than exploration*.

5 Related Work

The primacy bias in deep RL is intimately related to memorization, optimization in RL, and cognitive science. Various aspects of our work have been studied in the literature.

Addressing overfitting in RL. [11, 12] show that an approximator for value function gradually loses its expressivity due to bootstrapping which might amplify the effects of the primacy bias. [8] uses an on-policy buffer-free algorithm and distills the previous network after resetting it to improve generalization. Forms of non-uniform sampling including re-weighting recent samples [22] and prioritized experience replay (PER) [16] can be seen as a way to mitigate the primacy bias. SPR, which already uses PER as a component, still benefits from resets.

Forgetting mechanisms. In contrast to the well-known phenomenon of catastrophic forgetting [5], several works have observed catastrophic memorization [15], similar to the primacy bias. [3] notices higher sensitivity of the trained networks to early data. Resetting subnetworks has recently received more attention in supervised learning. [23] shows that forgetting might improve generalization and draws a connection to the emergence of compositional representations. These works complement the evidence about the primacy bias in deep RL and add to our analysis of the regularizing effect of resets.

Cognitive science. The primacy bias (also known as the *primacy effect*) has been studied in human learning for many decades [13]. [18] argues that outcomes of the first experience have a substantial and lasting effect on subsequent behavior and affect the outcomes of future decision making. Even though humans and RL systems learn under different conditions, our findings provide evidence that artificial agents also exhibit this type of bias.

6 Conclusion

This work identifies the primacy bias in deep RL, a damaging tendency of artificial agents to overfit early experiences. We demonstrate the dangers associated with this form of overfitting and propose a simple solution based on resetting a part of the agent. The experimental evidence across domains and algorithms suggests that resetting is an effective and generally applicable technique. We are intrigued by the results: if something as simple as resetting drastically improves the performance, a room for advancements in deep RL is enormous. Overall, this work sheds light on the learning process of deep RL agents, unlocks training regimes that were unavailable without resets, and opens possibilities for further studies improving both understanding and performance of deep reinforcement learning algorithms.

References

[1] Rishabh Agarwal et al. “Deep reinforcement learning at the edge of the statistical precipice”. In: *NeurIPS*. 2021.

[2] Marc G Bellemare et al. “The arcade learning environment: An evaluation platform for general agents”. In: *Journal of Artificial Intelligence Research* 47 (2013), pp. 253–279.

[3] Dumitru Erhan et al. “Why does unsupervised pre-training help deep learning?” In: *AISTATS*. 2010.

[4] William Fedus et al. “Revisiting fundamentals of experience replay”. In: *ICML*. PMLR. 2020, pp. 3061–3071.

[5] Robert M French. “Catastrophic forgetting in connectionist networks”. In: *Trends in cognitive sciences* 3.4 (1999), pp. 128–135.

[6] Tuomas Haarnoja et al. “Soft actor-critic algorithms and applications”. In: *arXiv preprint arXiv:1812.05905* (2018).

[7] Hado P van Hasselt et al. “When to use parametric models in reinforcement learning?” In: *NeurIPS*. 2019.

[8] Maximilian Igl et al. “Transient Non-stationarity and Generalisation in Deep Reinforcement Learning”. In: *ICLR*. 2021.

[9] Łukasz Kaiser et al. “Model Based Reinforcement Learning for Atari”. In: *ICLR*. 2019.

[10] Ilya Kostrikov et al. “Image augmentation is all you need: Regularizing deep reinforcement learning from pixels”. In: *ICLR*. 2021.

[11] Aviral Kumar et al. “Implicit Under-Parameterization Inhibits Data-Efficient Deep Reinforcement Learning”. In: *ICLR*. 2020.

[12] Clare Lyle et al. “Understanding and Preventing Capacity Loss in Reinforcement Learning”. In: *ICLR*. 2022.

[13] Philip H Marshall and Pamela R Werder. “The effects of the elimination of rehearsal on primacy and recency”. In: *Journal of Verbal Learning and Verbal Behavior* 11.5 (1972), pp. 649–653.

[14] Volodymyr Mnih et al. “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540 (2015), pp. 529–533.

[15] Anthony Robins. “Catastrophic forgetting in neural networks: the role of rehearsal mechanisms”. In: *Proceedings 1993 The First New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems*. IEEE. 1993, pp. 65–68.

[16] Tom Schaul et al. “Prioritized Experience Replay”. In: *ICLR*. 2016.

[17] Max Schwarzer et al. “Data-Efficient Reinforcement Learning with Self-Predictive Representations”. In: *ICLR*. 2020.

[18] Hanan Shteingart et al. “The role of first impression in operant learning.” In: *Journal of Experimental Psychology: General* 142.2 (2013), p. 476.

[19] Richard S Sutton. “Learning to predict by the methods of temporal differences”. In: *Machine learning* 3.1 (1988), pp. 9–44.

[20] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[21] Yuval Tassa et al. *dm_control: Software and Tasks for Continuous Control*. 2020. arXiv: `2006.12983 [cs.RO]`.

[22] Che Wang et al. “Striving for simplicity and performance in off-policy DRL: Output normalization and non-uniform sampling”. In: *ICML*. PMLR. 2020, pp. 10070–10080.

[23] Hattie Zhou et al. “Fortuitous Forgetting in Connectionist Networks”. In: *ICLR*. 2022.