Democratizing RL Research by Reusing Prior Computation

Due to the generality of reinforcement learning (RL), the prevailing trend in deep RL research is to develop agents that can efficiently learn *tabula rasa*, that is without any existing knowledge including prior computational work such as existing datasets, learned policies, LLMs *etc*. Unfortunately, the inefficiency of tabula rasa RL typically excludes the majority of the RL community outside certain resource-rich labs from tackling computationally-demanding problems. For example, the quintessential benchmark of training a typical deep RL agent on 50 Atari games in Arcade Learning Environment (ALE) [6] for 200M frames costs 1000+ GPU days, excluding all but a handful of labs [22]. Furthermore, while learning tabula rasa works well for small-scale research domains such as Atari or MuJoCo, it is the exception rather than the norm for solving larger-scale problems [*e.g.,* 3, 4, 9, 19, 20, 24, 25]. Large-scale RL systems often need to function for long periods of time and continually experience new data; restarting them from scratch to incorporate system or design (*e.g.,* algorithmic or architectural changes) may require weeks if not months of computation, and there may be millions of data points to re-process – this makes the tabula rasa approach impractical. Thus, as deep RL research move towards more complex and challenging benchmarks, the computational barrier to entry in RL research would be even substantially higher.

To address the inefficiencies of tabula rasa RL and unlock deep RL research for the masses, we present *reincarnating* RL (RRL) as an alternative research workflow or a class of problems the RL community should focus on. RRL seeks to maximally leverage prior computational work, such as learned network weights and collected data, to accelerate training across design iterations of an RL agent or when moving from one agent to another. In RRL, agents need not be trained tabula rasa, except for initial forays into new problems. For example, imagine a researcher who has trained an agent $\mathcal{A}_1$ for a long time (*e.g.,* weeks), but now this or another researcher wants to experiment with better architectures or RL algorithms. While the tabula rasa workflow requires re-training another agent from scratch, reincarnating RL provides the more viable option of transferring $\mathcal{A}_1$ to another agent and training this agent further, or simply fine-tuning $\mathcal{A}_1$. While areas such as offline RL, imitation learning, transfer in RL *etc* focus on developing methods to leverage prior computation, contrary to RRL, such areas don't strive to change how we do RL research by incorporating such methods as a part of our workflow. For example, we still mostly train ALE agents from scratch.

RRL can democratize research by allowing the broader community to tackle complex RL problems without requiring excessive computational resources. As a consequence, RRL can also help avoid the risk of researchers overfitting to conclusions from small-scale RL problems. Furthermore, RRL can enable a benchmarking paradigm where researchers continually improve and update existing trained agents, especially on problems where improving performance has real-world impact (*e.g.,* balloon navigation [7], chip design [20], tokamak control [12]). Furthermore, a common real-world RL use case will likely be in scenarios where prior computational work is available (*e.g.,* existing deployed RL policies), making RRL important to study. However, beyond some *ad hoc* large-scale reincarnation efforts with limited applicability (existing approaches can not be used for RRL when switching to arbitrary architectures or transferring policies to value / model-based agents), the community has not focused much on studying reincarnating RL as a research problem in its own right. To this end, we argue for developing general-purpose RRL approaches as opposed to prior *ad hoc* solutions.

Different RRL problems can be instantiated depending on how the prior computational work is provided: logged datasets, learned policies, pretrained models, representations, *etc*. As a step towards developing broadly applicable RRL approaches, we present a case study on how reincarnating RL can democratize research on ALE, one of the most widely studied RL benchmark. We'll discuss several avenues for future work such as enabling workflows that can incorporate knowledge provided in a form other than a policy, such as pretrained models, representations or LLMs, developing better methods for PVRL, and being able to utilize several sources of prior computation (multiple suboptimal teachers, LLMs etc). Furthermore, we believe that reincarnating RL would be especially important for continued progress in building embodied agents in open-ended domains [5]. As Newton put it "If I have seen further it is by standing on the shoulders of giants", we argue that reincarnating RL can substantially accelerate progress in deep RL by building on prior computational work, as opposed to always redoing this work from scratch.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

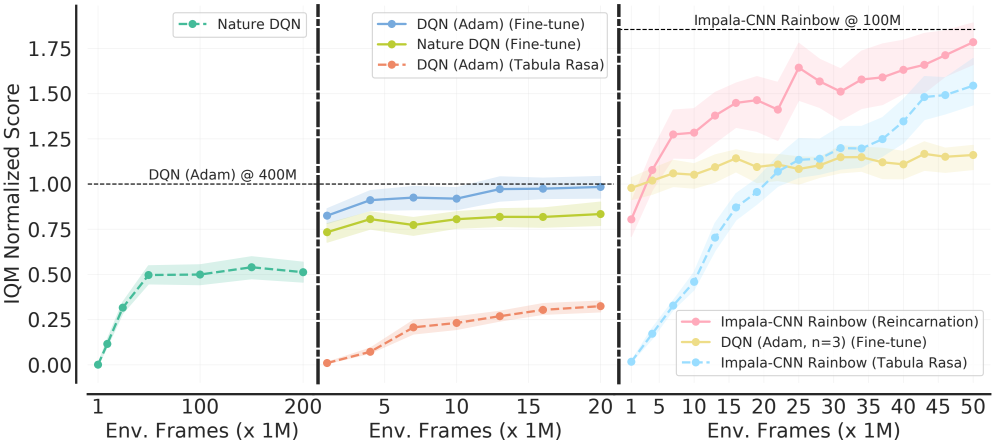

Figure 1: **A reincarnating RL workflow on ALE**. The plots show Interquartile mean [2] normalized scores over training, computed using 50 seeds, aggregated across 10 Atari games. The vertical separators correspond to loading network weights and replay buffer for fine-tuning while offline pre-training on replay buffer using QDagger for reincarnation. Shaded regions show 95% confidence intervals. We assign a score of 1 to DQN (Adam) trained for 400M frames and 0 to a random agent. **(Panel 1)** *Tabula rasa* Nature DQN [21] nearly converges in performance after training for 200M environment frames. **(Panel 2)** Reincarnation via fine-tuning Nature DQN with a reduced learning rate leads to 50% higher IQM with only 1M additional frames (leftmost point). Furthermore, fine-tuning Nature DQN while switching from RMSProp to Adam matches the performance of DQN (Adam) trained from scratch for 400M frames, using only 20M frames. **(Panel 3)** A modern ResNet (Impala-CNN [14]) with a better algorithm (Rainbow [17]) outperforms further fine-tuning n-step DQN. Reincarnated Impala-CNN Rainbow outperforms tabula rasa Impala-CNN Rainbow throughout training and requires only 50M frames to nearly match its performance at 100M frames. See Section .

Case Study on ALE: Democratizing Research via Reincarnating RL

As Mnih et al. [21]'s development of Nature DQN established the tabula rasa workflow on ALE, we demonstrate how iterating on ALE agents' design can be significantly accelerated using a reincarnating RL workflow, starting from Nature DQN, in Figure 1. For each game, training Nature DQN requires about 3-5 days on a single GPU using the Dopamine library [10]. Although Nature DQN used RMSProp, Adam yields better performance than RMSProp [1, 22]. While we can train another DQN agent from scratch with Adam, fine-tuning Nature DQN with Adam and 3-step returns, with a reduced learning rate, matches the performance of this tabula rasa DQN trained for 400M frames, using a 20 *times* smaller sample and compute budget (Panel 2 in Figure 1). As such, on a P100 GPU, fine-tuning only requires *training for a few hours rather than a week* needed for tabula rasa RL. Given this fine-tuned DQN, fine-tuning it further results in diminishing returns with additional frames due to being constrained to use the 3-layer convolutional neural network (CNN) with the DQN algorithm.

Let us now consider how one might use a more general reincarnation approach to improve on fine-tuning, by leveraging architectural and algorithmic advances since DQN, without the sample complexity of training from scratch (Panel 3 in Figure 1). Specifically, using QDagger(̃a specific algorithm for reincarnating RL) to transfer the fine-tuned DQN, we reincarnate Impala-CNN Rainbow that combines Dopamine Rainbow [17], which incorporates distributional RL [8], prioritized replay [23] and n-step returns, with an Impala-CNN architecture [14], a deep ResNet with 15 convolutional layers. Tabula rasa Impala-CNN Rainbow outperforms fine-tuning DQN further within 25M frames. Reincarnated Impala-CNN Rainbow quickly outperforms its teacher policy within 5M frames and maintains superior performance over its tabula rasa counterpart throughout training for 50M frames. To catch up with the performance of this reincarnated agent's performance, the tabula rasa Impala-CNN Rainbow requires additional training for 50M frames (48 hours on a P100 GPU). Overall, these results indicate how past research on ALE could have been accelerated by incorporating a reincarnating RL approach to designing agents, instead of always re-training agents from scratch.

Takeaway: Reincarnating RL could positively impact society by reducing the computational burden on researchers and is more environment friendly than tabula rasa RL. For example, reincarnating RL allow researchers to train super-human Atari agents on a single GPU within a span of few hours as opposed to training for a few days. Additionally, reincarnating RL is more accessible to the wider research community, as researchers without sufficient compute resources can build on prior computational work from resource-rich groups, and even improve upon them using limited resources. Furthermore, this democratization could directly improve RL applicability for practical applications, as most businesses that could benefit from RL often cannot afford the expertise to design in-house solutions.

Reproducibility, Comparisons and Generalizability

Scientific Comparisons. Fairly comparing reincarnation approaches entails using the exactly same computational work and workflow. To enable this, it would be beneficial if the researchers can release model checkpoints and the data generated (at least the final replay buffers), in addition to open-source code for their trained RL agents. Indeed, to allow others to use the same reincarnation setup as our work, we have already open-sourced DQN (Adam) agent checkpoints and the final replay buffer at `gs://rl_checkpoints`.

Generalizability. The generalizable findings in reincarnating RL would be about comparing algorithmic efficacy given access to existing computational work on a task. As such, the performance ranking of reincarnation algorithms is likely to remain consistent across different teachers. Practitioners can use the findings from reincarnating RL to try to improve on an existing deployed RL policy (as opposed to being restricted to running tabula rasa RL).

Reproducibility. Reproducibility *from scratch* is challenging in RRL as it would require details of the generation of the prior computational work (*e.g.,* teacher policies), which may itself has been obtained via reincarnating RL. As reproducibility from scratch involves reproducing existing computational work, it could be more expensive than training tabula rasa, which beats the purpose of doing reincarnation. Furthermore, reproducibility from scratch is also difficult in NLP and computed vision, where existing pretrained models (*e.g.,*, GPT-3) are rarely, if ever, reproduced / re-trained from scratch but almost always used as-is. Despite this difficulty, *pretraining-and-fine-tuning* is a dominant paradigm in NLP and vision [*e.g.,* 11, 13, 16, 18], and we believe that a similar difficulty in RRL should not prevent researchers from investigating and studying this important class of problems. Instead, we expect that RRL research would build on **open-sourced** prior computational work. Akin to NLP and vision, where typically a small set of pretrained models are used in research, we believe that research on developing better reincarnating RL methods can also possibly converge to a small set of open-sourced models / data on a given benchmark, *e.g.,* the agents and data we released on Atari or the $25,000$ trained Atari agents released by Gogianu et al. [15], concurrent to this work.

References

[1] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.

[2] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[4] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[5] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *arXiv preprint arXiv:2206.11795*, 2022.

[6] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

[7] Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.

[8] Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2022. http://www.distributional-rl.org.

[9] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[10] Pablo Samuel Castro, Subhodeep Moitra, Carles Gelada, Saurabh Kumar, and Marc G Bellemare. Dopamine: A research framework for deep reinforcement learning. *arXiv preprint arXiv:1812.06110*, 2018.

[11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.

[12] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pages 1407–1416. PMLR, 2018.

[15] Florin Gogianu, Tudor Berariu, Lucian Bușoniu, and Elena Burceanu. Atari agents, 2022. URL https://github.com/floringogianu/atari-agents.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[17] Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[18] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[19] Yao Lu, Karol Hausman, Yevgen Chebotar, Mengyuan Yan, Eric Jang, Alexander Herzog, Ted Xiao, Alex Irpan, Mohi Khansari, Dmitry Kalashnikov, et al. Aw-opt: Learning robotic skills with imitation andreinforcement at scale. In *Conference on Robot Learning*, pages 1078–1088. PMLR, 2022.

[20] Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nazi, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

[21] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[22] Johan S Obando-Ceron and Pablo Samuel Castro. Revisiting rainbow: Promoting more insightful and inclusive deep reinforcement learning research. In *International Conference on Machine Learning (ICML)*, 2021.

[23] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[24] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[25] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354, 2019.