Internship / PhD Position

RL↔CL: Connections between Reinforcement Learning and Continual Learning

Contact: Alex Davey — alex.davey@inria.fr

Abstract

Reinforcement Learning (RL) can be viewed not as a single learning problem, but as a sequence of related learning problems. As the policy improves, the induced data distribution and target values change, leading to a different learning "task". This sequence of learning problems shares many similarities with Continual Learning (CL), where one is presented with an explicit sequence of different tasks, and must perform well on the new task while retaining performance on the old ones. This project will investigate in what sense RL can be usefully formalised as a CL problem, and exploit connections between these two fields that have already been observed in the literature.

One of the three subtopics below will form the basis of a short internship (2-3 months) starting Jan 2027 (flexible) . If successful, this will lead to a three-year fully-funded PhD position.

Research team	Inria TAU
Laboratory	LISN
Institution	Inria Paris-Saclay
Doctoral school	Université Paris-Saclay
Location	Paris, France
Supervisor	Alex Davey
Co-supervisors	Guillaume Charpiat, Sylvain Chevallier
Expected start	Internship Jan 2027 (2-3 months); followed by PhD
PhD Duration	3 years
Funding	PEPR SAIF

Introduction

In Reinforcement Learning (RL), we typically start with some initial random policy π_0, and learn a sequence of policies $\pi_0, \pi_1, \ldots$ that we hope increasingly approximates the optimal policy $\pi^\star$. Unlike in supervised learning, these updated policies induce a new state visitation distribution, and new value targets to learn [1]. This is true even in a single-task (i.e. stationary) Markov Decision Process (MDP), where the reward and transition functions are fixed.

In Continual Learning (CL), a learner is presented with a sequence of tasks $\mathcal{T}_1, \mathcal{T}_2, \ldots$, and is expected to acquire the new task while retaining performance on the earlier tasks, a tension sometimes termed the stability-plasticity dilemma [2]. In RL, by contrast, we are normally interested only in maximising performance on the latest task. To make the connection precise, we must therefore define what should be remembered about previous learning problems or, equivalently, *what to forget*. However, these connections have mostly been drawn as analogies between specific algorithms, rather than as a formal connection. This PhD will both investigate how single-task RL can be formalised as a CL problem, and exploit known connections between the two fields, to improve the robustness of RL algorithms in the presence of function approximation errors. There are three main categories of Continual Learning algorithm [2]:

1. **Regularisation methods**, which penalise changes in parameter space deemed important to earlier tasks, analogous to regularised policy iteration methods in RL such as TRPO [3].
2. **Rehearsal methods**, which store a small subset of data associated with each task, analogous to the use of replay buffers in RL.
3. **Parameter Isolation methods**, where some proportion of the parameters of a network are dedicated to each task [4]. This is less explored in RL, but is closest to growing neural networks in Deep RL.

This project will focus initially on single-task RL (through the viewpoint of CL), but Continual Learning and Continual Reinforcement Learning are natural extensions. This project is funded by the PEPR SAIF [5], which aims to promote the use of formal methods to improve the safety, reliability, and explainability of machine learning systems.

Research problem

The central question is: **in what sense can Reinforcement Learning be usefully formalised as a Continual Learning problem?** To make this more concrete, we can consider e.g. (generalised) policy iteration [1], which encompasses a wide variety of RL algorithm. We start with an initial policy π_0 and alternate between policy evaluation and policy improvement:

$$\pi_0 \xrightarrow{\text{eval.}} V^{\pi_0} \xrightarrow{\text{imp.}} \pi_1 \xrightarrow{\text{eval.}} \cdots \xrightarrow{\text{imp.}} \pi^\star \xrightarrow{\text{eval.}} V^\star \tag{1}$$

where at each policy evaluation step we find the fixed point $V^{\pi_k} = T^{\pi_k} V^{\pi_k}$ of the Bellman operator T^{π_k}, and at each policy improvement step we improve the policy at each state s to find π_{k+1}.

In exact policy iteration, the policy improvement theorem guarantees a monotonic improvement $V^{\pi_{k+1}}(s) \geq V^{\pi_k}(s)$ for all states [1]. To understand why this is indeed analogous to Continual Learning and not simply a sequential learning problem, we have to introduce function approximation. Then, we do not update our value function and policy on all states, but only on a subset $\mathcal{D}_k$ of states (often derived from the previous policy π_{k-1}), and with function approximation these updates also change the function outside $\mathcal{D}_k$. This motivates the notion of a "task": learn a policy π which performs well on the empirical distribution of $\mathcal{D}_k$. Indeed, in the case where the supports of the distributions overlap little, learning one task should ideally not degrade performance on the other, yet with a shared function approximator it generally does.

The three classes of Continual Learning method provide a framework for the research programme.

1. Regularisation methods

In CL, regularisation methods penalise changes in parameter space deemed important for earlier tasks, e.g. as in Elastic Weight Consolidation (EWC) [6], resulting in a Fisher-weighted penalty. This is analogous to trust-region approaches such as TRPO [3] and more generally to regularised policy iteration methods, which constrain changes in the function space with respect to the previous policy.

However, in general this regularisation is only with respect to the last policy π_{k-1}, rather than all previous policies. In exact tabular policy iteration this is well-motivated due to the policy improvement theorem mentioned previously: all information from previous policies is irrelevant or already incorporated, but this does not necessarily hold in the case of function approximation. This research direction considers regularisation with respect to all previous policies, rather than just the latest one, informed by an estimate of the errors associated to each approximate Q-function.

2. Rehearsal methods

In CL, rehearsal methods revisit a subset of data from previous tasks [2]. In off-policy RL, this is often implemented by a large replay buffer which stores transitions collected by earlier policies. This naive first-in-first-out buffer works well in practice but is inefficient and does not scale well to very long-running tasks or memory-contrained systems.

In supervised learning, coresets [7] are weighted, compressed subsets $\mathcal{S} \subset \mathcal{B}$ whose gradient approximates the full gradient of the loss $\nabla \ell$ on the original dataset $\mathcal{B}$:

$$\max_{\theta} \left\| \sum_{x \in \mathcal{B}} \nabla \ell(\theta, x) - \sum_{x \in \mathcal{S}} \lambda_x \nabla \ell(\theta, x) \right\| \leq \varepsilon, \tag{2}$$

for which one can prove that, for convex losses, gradient descent on $\mathcal{S}$ converges to a near-optimal solution at the same rate as on $\mathcal{B}$ [7]. In RL this differs, because the data arrives in a streaming fashion, and Temporal Difference (TD) updates have moving targets, but also because the data encodes the dynamical structure of the MDP. This direction would introduce a coreset approach which explicitly accounts the MDP structure.

3. Parameter Isolation methods

In CL, parameter isolation methods dedicate a subset of parameters to each task [2,4]. The analogue of this is relatively understudied in Deep RL; however, dedicating a separate network to each Q-function has shown some promise [8]. A more efficient way to do this is via Neural Network Growth, which has been successfully used to reduce plasticity loss [9,10]. However, these methods are heuristic and do not propose a principled criterion for where or when to grow, unlike supervised learning methods based on expressivity bottlenecks [11], or earlier RL works based on the Bellman residual [12,13].

Research environment

The candidate will join the Inria TAU team, specifically in the Frugal subteam, at the Laboratoire Interdisciplinaire des Sciences du Numérique (LISN). The doctorate will be associated with Université Paris-Saclay. This project will be supervised by Alex Davey, with co-supervision by Guillaume Charpiat and Sylvain Chevallier.

The Frugal subteam of TAU works primarily on Growing Neural Networks, but is more generally interested in optimisation and natural gradient methods. There is also a growing part of the team interested in Reinforcement Learning, and the wider TAU team has expertise in RL for control systems.

Candidate profile

The following recent papers by the group, StaQ [8] and TINY [11], may provide an idea of the kind of work involved in this project.

Essential: Applicants should hold, or be close to completing, a relevant Master's degree or equivalent qualification, with solid foundations in linear algebra, probability, optimisation, and a good understanding of deep learning.

Desirable: Programming experience, particularly Python and PyTorch, and familiarity with Reinforcement Learning or Continual Learning would be useful. Experience with proofs, error bounds or formal methods is also welcome. The position suits someone interested in developing both mathematical and experimental research skills.

How to apply

Please send a CV to alex.davey@inria.fr. We strongly recommend sending a copy of your Master's thesis (or other relevant academic work) as well. Applications will be considered until the position is filled.

References

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd edn (MIT Press, 2018).

[2] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, A Continual Learning Survey: Defying Forgetting in Classification Tasks, IEEE Transactions on Pattern Analysis and Machine Intelligence **44**, 3366 (2022).

[3] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, *Trust Region Policy Optimization*, in *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37 (2015), pp. 1889–1897.

[4] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, Progressive Neural Networks, arXiv preprint arXiv:1606.04671 (2016).

[5] PEPR IA, *SAIF: Sûreté de l'intelligence Artificielle Investiguée par les méthodes Formelles*, https://www.pepr-ia.fr/projet/saif/.

[6] J. Kirkpatrick et al., Overcoming Catastrophic Forgetting in Neural Networks, Proceedings of the National Academy of Sciences **114**, 3521 (2017).

[7] B. Mirzasoleiman, J. Bilmes, and J. Leskovec, *Coresets for Data-Efficient Training of Machine Learning Models*, in *Proceedings of the 37th International Conference on Machine Learning*, Vol. 119 (2020), pp. 6950–6960.

[8] A. Davey, A. Shilova, B. Driss, and R. Akrour, StaQ: a Finite Memory Approach to Discrete Action Policy Mirror Descent, Reinforcement Learning Journal (2026).

[9] Z. Kang, C. Hu, Y. Luo, Z. Yuan, R. Zheng, and H. Xu, *A Forget-and-Grow Strategy for Deep Reinforcement Learning Scaling in Continuous Control*, in *Proceedings of the 42nd International Conference on Machine Learning*, Vol. 267 (2025), pp. 28921–28942.

[10] J. Liu, J. Obando-Ceron, A. Courville, and L. Pan, *Neuroplastic Expansion in Deep Reinforcement Learning*, in *International Conference on Learning Representations* (2025).

[11] M. Verbockhaven, T. Rudkiewicz, S. Chevallier, and G. Charpiat, Growing Tiny Networks: Spotting Expressivity Bottlenecks and Fixing Them Optimally, Transactions on Machine Learning Research (2024).

[12] M. Milani Fard, Y. Grinberg, A.-M. Farahmand, J. Pineau, and D. Precup, *Bellman Error Based Feature Generation Using Random Projections on Sparse Spaces*, in *Advances in Neural Information Processing Systems*, Vol. 26 (2013).

[13] S. Tosatto, M. Pirotta, C. D'Eramo, and M. Restelli, *Boosted Fitted Q-Iteration*, in *Proceedings of the 34th International Conference on Machine Learning*, Vol. 70 (2017), pp. 3434–3443.