Low-level Algorithms for Continual and Meta Reinforcement Learning (Summary)

The reinforcement learning (RL) approach to artificial intelligence has had many impressive successes, including superhuman performance in Backgammon, Atari games, Go, Chess, Poker, StarCraft, and simulated race-car driving. The RL approach is special in that it treats the entirety of an autonomous intelligent agent interacting with its environment. Even simple RL agents sense their environment, take action, and have a goal (maximizing reward). More advanced agents integrate perception and planning into their decision-making.

Despite the successes of RL, its low-level learning algorithms are limited in *continual learning* settings. The most impressive successes of RL use artificial neural networks, but these deep reinforcement learning (DRL) methods have great difficulty continuing to learn as they continue to interact with their environment. The standard strategy in DRL is to maintain a buffer of past experience and selectively replay samples from it. We propose instead to refine the low-level learning algorithms so that they themselves are capable of continual learning and the complexity of replay buffers is not needed. In initial work, we have shown loss of plasticity in standard supervised learning domains (ImageNet and MNIST) and developed simple variants of the backpropagation algorithm that maintain plasticity indefinitely.

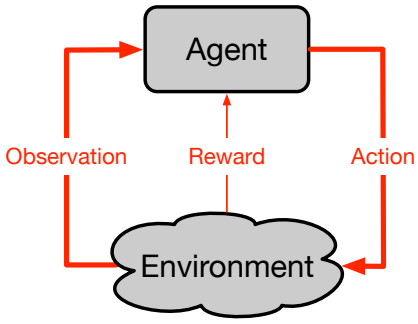
An agent that continues to learn will gain extensive experience with the process of learning; it could potentially learn to learn better, a phenomena called *meta-learning*. A meta-learning agent may learn to generalize better from state to state by forming better state representations. The standard approach to meta-learning today involves transfer between tasks, but we propose to instead use a single task that is too large to be solved exactly (the under-parameterized, or big-world case), or to emulate this with a single nonstationary task. In these cases, we changes to the low-level algorithms may be sufficient to obtain meta-learning. In particular, we are interested in meta-gradient algorithms for step-size adaptation such as Incremental Delta-Bar-Delta (IDBD). We expect to show that IDBD is qualitatively different from optimizers commonly used in modern DRL, such as Adam and RMSprop, in ways that make it a better starting place for developing low-level algorithms for representation learning.

Continual learning and meta-learning are closely related. Both directly contribute to the efficiency and robustness of ongoing learning processes. At the lowest level, we believe these two abilities are different aspects of the same algorithmic mechanisms.

To the extent that this research is successful, it will contribute to a better understanding of the domain-independent algorithms and principles of intelligence. In the near term, the developed technology will improve the ability of our machine learning systems to learn efficiently in applications that involve change and nonstationarity, which is arguably almost all of them.

Low-level Algorithms for Continual and Meta Reinforcement Learning (Proposal)

The field of artificial intelligence (AI) seeks to understand the principles of intelligence well enough to create it through technology, where intelligence may be defined as "the computational part of the ability to achieve goals in the world" (McCarthy 1997). The reinforcement learning (RL) approach to AI, which we pursue in the proposed research, is consistent with this definition: "in the world" is interpreted as meaning in interaction with an external environment, and "achieve goals" is operationalized as obtain *reward*, where reward is a special scalar signal received from the environment. In RL, as in natural intelligence, the intelligent agent has a temporal existence; it processes sensory signals (observations) over time and uses them to make decisions and generate motor signals (actions) over time. Intelligence is then the ability to a maximize reward, summed over time, by taking actions informed by observations. This perspective is summarized in the standard RL diagram shown on the right. The overall objective of research on the RL approach to AI, such as that in my laboratory, is to design the agent's algorithms so that they obtain a lot of reward in a wide range of environments. RL algorithms have played important roles in many of the most important successes of AI, including in Go, Chess, Jeopardy!, Backgammon, Atari games, aerial acrobatics, Starcraft, and race-car driving.



An intelligent agent's computational resources may be vast, but of course they are not infinite, nor is the time available to compute each action. The environment likely includes many other agents of equal complexity, which implies that the environment is *much more complex* than the agent. From this it more or less directly follows that the agent's action selections cannot, in general, be optimal, and that its understanding of the world cannot, in general, be accurate in all its details. The agent must settle for approximate policies and, if it forms a model of the environment, a grossly approximate model. This is a problem in particular for agent algorithms (or theories) that assume that the agent's policies or models become exact in the limit of infinite data, but really it is a major challenge for all methods. I call it the *Big World Problem*. The full significance of the Big World Problem is only beginning to be recognized and accepted.

One consequence of the Big World Problem is that while it can be helpful for an agent designer to initialize their agent with knowledge of the environment, such *prior domain knowledge* is rarely sufficient. Today's AI agents are already much too complex for their designers to understand the details of their learned approximations. As the agents increase in complexity—while the environment remains even more complex—designers cannot know the best approximations just as they cannot know all the details of the particular environment their agent will be exposed to. As agent complexity increases, a larger fraction of the knowledge has to be learned by the agent rather than built in by the designer.

Continual Learning

A second consequence of the Big World Problem is that learning must often be *continual* (never ending). It is typical in a large, complex environment for the part of it being encountered to change over time. In such cases, the agent will do best if it adapts its approximations to the part being encountered. Because the agent is limited in its approximation abilities, its approximations must continually adapted to the current part. That is, it must continue to learn, just as natural learning systems do.

Unfortunately today's most advanced learning systems, those based on deep neural networks, do not work well in continual learning settings. One aspect of this is the well-known phenomena of *catastrophic forgetting*, the tendency of artificial neural networks to forget almost all of what they have previously learned when exposed to new data (McCloskey & Cohen 1989; French 1999; Kirkpatrick et al. 2017). Less attention has been paid to the equally important issue of losing the ability to learn new

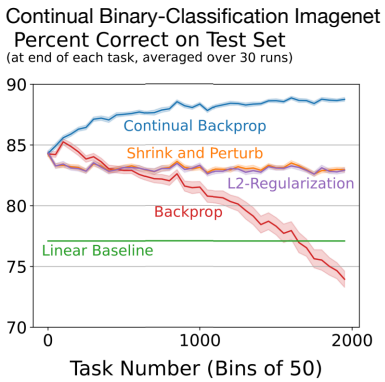
things, or *loss of plasticity*. Loss of plasticity was shown in early neural network experiments in the psychology literature (e.g., Ellis & Lambon 2000; Zevin & Seidenberg 2002), and recently in the machine learning literature (Ash & Adams 2020; Nishikin et al. 2022; Lyle et al. 2022). However, these prior works did not establish the phenomena definitively, either because they did not use modern deep learning methods, because of confounding due to the complexities of RL settings, or other issues. To definitively show the inadequacy of modern deep learning methods in continual learning settings is the first objective of this project's proposed research.

My interpretation is that modern deep learning systems have become too specialized to the setting in which training occurs once and then never again. These *train-once* learning methods were perfected for the classical supervised learning setting in which there is a single large training set. When applications to RL were explored (which inherently involve some continual learning), it was expedient to modify the train-once methods with work-arounds such as replay buffers rather than to develop new methods that could learn continually. Now, almost a decade later, machine learning researchers see the need to need for continual learning in order to handle large or changing environments (e.g., Parisi et al. 2019; Khetarpal et al. 2020) and are organizing specialized meeting to focus on it, such as the Conference on Life-long Learning Agents (CoLLAS). A primary objective of the project's research is to develop new methods for continual learning.

Recent Progress in Continual Learning

Work in my lab has been exploring approaches to continual learning based on selective re-initialization of hidden units in neural network. The ideas date back to the notions of random representations (Sutton & Whitehead 1993) and representation search (Mahmood & Sutton 2013; Mahmood 2017). Two recent MSc theses in my lab (Dohare 2020; Rahman 2020) developed and tested an algorithm, *continual backpropagation*, for maintaining plasticity in continual learning. Continual backpropagation is the same as the conventional backpropagation algorithm except that hidden units are reinitialized to small random weights throughout training (instead of only once at the beginning of training). The reinitialized units are few and carefully selected to interfere minimally with the ongoing behavior of the network. The overall effect is a form of generate-and-test search for good hidden units. The small random weights provide diversity, generating many seeds that may develop into useful features. Those that are found useful are preserved, and those that aren't are eventually re-initialized to try again. We have shown that this simple change to backpropagation can sometimes dramatically improve its performance on continual learning problems. The best way of measuring unit utility is one of the questions still being explored.

In current work in my lab, we have been performing extensive experiments on continual-learning versions of supervised learning domains such as ImageNet (Russakovsky et al. 2015) and MNIST (LeCun et al. 2010). In ImageNet, for example, the 1000 classes are taken in pairs to produce binary classification tasks that a single network encounters in sequence. The performance of backpropagation often improves over the first few tasks, but in the long run it loses plasticity and cannot adapt to new tasks, whereas continual backpropagation continues to do well. This is extensive and systematic empirical research that needs be done with care and statistical rigour in order to be convincing. The plot at right shows the kind of results we are getting on ImageNet for various algorithms as a function of task number in a sequence of 2000 binary classification tasks. To complete and publish this work prominently is an early objective of the project.



Meta-gradient Methods for Meta Learning

We turn now to the second major focus of the research project: low-level algorithms for meta learning.

When learning is continual, the agent gets repeated experience with learning, and over time it is possible for it to become better at learning, a process known as *meta learning*. Typically meta learning is used to adapt the parameters of a learning algorithm, such as step sizes, initial weights, or representational weightings. Meta learning of step-size parameters on a per-weight basis is the second focus of the project's research. It is natural to study continual and meta learning together because they arise in the same settings and can involve some of the same algorithmic machinery. In particular, features that have meta learned to have large step sizes on their outgoing weights can be taken to have high utility to the network even if those weights are small.

Meta-learning has been explored extensively within machine learning for many years (e.g., see Thrun & Pratt 1998). A class of meta-learning methods that appears particularly powerful and that has attracted considerable attention recently are those based on stochastic gradient descent or ascent (e.g., Andrychowicz et al. 2016; Finn, Abeel & Levine 2017; Xu, van Hasselt & Silver 2018). Almost all such *meta-gradient methods* use a gradient method in the base learning system as well. Meta-gradient methods have shown promise; they are very general and in some cases have achieved learning performance equal to that for hand-tuned meta-parameters.

The earliest meta-gradient methods were for setting step sizes. Sutton (1981) devised meta-gradient algorithms for meta-learning the step size of a servomechanism. Jacobs (1988), Sutton (1992), and Schraudolph (1999; 2002) extended this work to meta-learn many step-size parameters, one for each weight of a multi-layer neural network. The ideas underlying my 1992 algorithm, called Incremental Delta-Bar-Delta, or *IDBD*, were originally and contemporaneously developed as models of biological meta-learning systems (Sutton 1982; Gluck, Glauthier & Sutton 1992; Schweighofer & Arbib 1998). Schraudolph's method, called called Stochastic Meta Descent, or *SMD*, was used in many applications (see Sutton 2022). A limitation of SMD and IDBD is that they have some parameters of their own (meta-meta-parameters); Mahmood et al. (2012) devised a more robust version of IDBD that removed the need to tune any parameters manually. Koop (2008) developed a form of IDBD for logistic rather than linear functions such that it is suited to classification rather than regression.

Recent Progress in Meta Learning

Recent work on meta learning in my lab has focused on step sizes and the linear, non-stationary case as in the original work on IDBD but, unlike in that original work, now we are concerned with the ability to meta-learn *efficiently* rather than just to meta-learn at all. We are exploring several forms of normalization that have large effects on meta learning's efficiency. One is just normalization of the input signals to the linear unit; if these are translated to have mean zero and scaled to have unit variance it can dramatically accelerate both learning and meta learning. We also have some evidence that limiting the largest absolute value of the signals improves learning speed, which would imply two-valued features are to be preferred over, say, Gaussian-distributed signals.

A second form of normalization is to the meta-gradients themselves, which are sensitive to the scale of the target signals. To understand this, first note that the base learning process (that using the gradient of the squared error with respect to the weights) is not sensitive to the size of the targets in a supervised regression task. If the targets are ten times larger, then the errors and weight changes are also ten times larger. In particular, if the step sizes are such that the error is reduced by half with the original targets, then it will still be reduced by exactly half with the ten-times-larger targets. Unfortunately, the same is not true at the meta level. If the targets are enlarged, then the meta-gradients are enlarged, and the meta-step size that worked well for the original targets will have to be changed for the ten-times-larger targets. Or, as we are exploring, it appears possible to measure the magnitude of the meta-gradients and then normalize them so that the same meta-step-size parameter can be used even it the magnitude of the targets change. This work is a continuation of that by Rupam Mahmood (cited above) when he was a student in my lab.

Of course, there are many existing step-size adaptation methods in the literature other than IDBD, such as Adam (Kingma & Ba 2014) and RMSprop (Hinton et al. 2012). These are really in a different

category from algorithms like IDBD that are based on meta-gradient descent, and have a very different performance profile. They are normalization methods not unlike the normalizations discussed in the previous two paragraphs that we are exploring as additions to IDBD. In that sense, the current work can be seen as a combination of IDBD and existing Adam/RMSprop ideas. In any event, a first point is that IDBD has strengths not provided by other step-size adaptation methods, for example, in adapting differentially to stochasticity and non-stationarity. We are preparing now a direct demonstrations to establish these qualitative differences.

Objectives

The near-term objectives of the continual learning part of the project are 1) to demonstrate loss of plasticity in ImageNet and MNIST for a wide rang of deep learning methods, 2) to show that continual backpropagation maintains plasticity 3) to publish a prominent journal article on loss of plasticity, and 4) to extend our unit-utility measure to recurrent networks. The near-term objectives for the meta-learning part of the project are 1) to design meta-learning algorithms based on IDBD that are more robustly efficient in adapting per-weight step sizes of a linear network using multiple normalizations, 2) to extend incremental step-size adaptation to general feed-forward and then recurrent networks, and 3) to publish a paper on the new methods and on how they provide fundamentally new capabilities beyond conventional neural network optimizers.

A medium-term objective is to unify the continual and meta learning aspects of the project. The unit-utility measure in continual learning should be based not just on learned weights but also on learned step sizes. If this integration is done well, we will achieve integrated non-linear learning and meta learning in neural networks. The networks will be capable of shaping their representations and altering the way they generalize without human input. This would be a step-change improvement in learning in neural networks and would be another transformative breakthrough in machine learning applications that can continually learn and improve the way they learn.

The long-term objective of the project is to produce improved low-level learning algorithms that will support ambitious model-based agent architectures for AI such as those outlined in the Alberta Plan (Sutton et al. 2022a,b).

Impact

Computationally efficient learning methods form the core of RL and modern AI. Many might believe that there is little more to be done with low-level methods, that they have already reached the limit of their refinement. I could not disagree more. As we seek fundamental principles of learning and intelligence, we should seek above all to strengthen the low-level algorithms. The better we understand these components, and the more we can do reliably and scaleably with them, and the more powerful the systems we will be able to build with them.

In particular, the need for continual learning is becoming widely recognized. The practice in today's applications is to use train-once methods and to re-train them from scratch each time a significant chunk of new data is obtained. But the training process is computationally intensive and requires some human participation; we can save expense and maybe the climate by training incrementally with continual learning algorithms that monitor themselves. I believe it is inevitable that today's train-once learning algorithms will be replaced almost completely with continual learning algorithms. Continual learning will be embedded in a new iteration of deep learning systems, and meta learning will be embedded in the iteration after that.

In the longer term, say over the next decade, improved low-level learning algorithms such as those to be developed in this project will be enabling of fundamental advances in AI. A long-standing goal in AI has been to learn and discover general world knowledge expressed as regularities in the agent's sensorimotor data stream (e.g., see Sutton 2009). Model-based RL architectures have been outlined for this, and sophisticated algorithms have been developed, such as for temporal abstraction and for off-policy and policy-gradient learning. Yet these learning methods have many rough edges and, in particular, they lose

plasticity, robustness, and the ability to shape generalization when they are scaled to larger networks. Moreover, model-based RL architectures have multiple components that learn simultaneously at different time scales, such as policies, value functions, transition models and state construction processes. With so many moving parts, train-once learning algorithms become a debilitating weakness that greatly limits a researchers ability to understand the system and make scientific progress. A better understanding of low-level continual and meta learning is needed for continued progress toward understanding intelligence and creating it with technology.

Methodology

Much of the research and algorithm development in my lab is conceptual and done on paper and whiteboard. Simple worlds are imagined as well as the corresponding behaviour of the algorithm. At a certain point we switch to formal analysis, leading ultimately to formal proofs of convergence or equivalences. We also make extensive use of computational "microworlds," small imaginary worlds that are completely understood as Markov decision processes and that can be used to test learning algorithms and to compare their performance. The members of my group get extensive training in the appropriate way to vary system features and algorithm parameters to permit fair comparisons.

Today it is important to sometimes use large domains and large networks. We need large computational resources both because the learned systems are much larger and we need to explore their many variations in a statistically reliable way. A laptop computer, even a powerful one, is generally not sufficient. One needs access to large computation servers. We have used Compute Canada extensively for this purpose and it has served well. By one year from now we will have access to larger dedicated computation servers through the Pan-Canadian AI program and the Alberta Machine Intelligence Institute.

Recent Progress with Respect to My Most-Recent Discovery-Grant Proposal

My previous discovery-grant proposal, from 2013, focused on the development of new algorithms for *temporal-difference (TD) learning*, a core technology at the heart of much of the excitement and many of the successes of modern reinforcement learning. The work on that project went exceedingly well and resulted in two new families of more powerful TD learning algorithms.

The first new family of TD learning algorithms was *true online TD learning* (Van Seijen & Sutton 2014, Van Seijen et al. 2016). TD learning methods can be viewed as incremental, computationally efficient ways of achieving an overall result. Often the overall result is easier to understand—such as approximating the expected sum of future rewards—but harder to learn incrementally from contemporaneously available information. Conventional TD methods solve this problem, but only approximately. The classical algorithm TD(λ) achieves a result that is only approximately equivalent to the desired overall result. For 26 years this was thought to be the best that was possible in a linear-complexity, independent-of-span algorithm, but in 2014, in my lab, Harm van Seijen developed *true online TD(λ)*, which obtained an exact equivalence with only a little extra linear computation. This work was published in ICML in 2014 and in JMLR in 2016. These two publications have been cited 210 times.

The second new family of TD learning algorithms to come the last project was *Emphatic-TD* learning (Sutton et al. 2016). Emphatic-TD methods are a new solution to the problem of the "the deadly triad," the tendency of off-policy learning method to be unstable when combined with TD learning and function approximation. This has been a known key problem since 1995. The only previously known solution (that retained linear computational complexity and converged to the right parameter) was Gradient-TD methods proposed by lab (Sutton et al. 2009, Maei 2011). Emphatic-TD methods improved over Gradient-TD methods in that they have only one modifiable weight vector and one step-size parameter. Later we were able to show empirically that they are also superior in other ways important in practice. The main Emphatic-TD paper has been cited 207 times.

Both true-online-TD and Emphatic-TD learning methods have been influential around the world as well as in later work in our lab. These are the best TD and off-policy algorithms known, and can be expected to play important roles in the future of RL and AI.

References

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., De Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems,* pp. 3981-3989.

Ash, J., Adams, R.P. (2020). On warm-starting neural network training. *Advances in Neural Information Processing Systems*. pp. 3884–3894.

Dohare, S. (2020). *The Interplay of Search and Gradient Descent in Semi-stationary Learning Problems*. University of Alberta MSc thesis.

Ellis, A. W., Lambon-Ralph, M. A. (2000). Age of acquisition effects in adult lexical processing reflect loss of plasticity in maturing systems: Insights from connectionist networks. *Journal of Experimental Psychology: Learning, Memory, and Cognition 26*(5):1103-1123.

Finn, C., Abbeel, P., Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning* (pp. 1126–1135).

French, R. M. (1999). Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences 3:*128–135.

Gluck, M., Glauthier, P., Sutton, R. S. (1992). Adaptation of cue-specific learning rates in network models of human category learning, *Conference of the Cognitive Science Society*, pp. 540–545, Erlbaum.

Hinton, G., Srivastava, N., Swersky, K. (2012). Neural networks for machine learning, lecture 6a, overview of mini-batch gradient descent.

Jacobs, R. A. (1988). Increased rates of convergence through learning rate adaptation. *Neural Networks 1*(4):295–307.

Khetarpal, K., Riemer, M., Rish, I., Precup, D. (2020). Towards continual reinforcement learning: A review and perspectives. arXiv:2012.13490.

Kingma, D. P., Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., GrabskaBarwinska, A., Hassabis, D., Clopath, C., Kumaran, D., Hadsell, R., 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences 114*:3521–3526.

Koop, A. (2008). *Investigating experience: Temporal Coherence and Empirical Knowledge Representation*. University of Alberta MSc thesis.

LeCun, Y., Cortes, C., Burges, C. (2010). MNIST handwritten digit database.

Lyle, C., Rowland, M., Dabney, W. (2022). Understanding and preventing capacity loss in reinforcement learning. *International Conference on Learning Representations*.

Maei, H. R. (2011). *Gradient Temporal-Difference Learning Algorithms*. University of Alberta PhD thesis.

Mahmood, A. (2017). *Incremental off-policy reinforcement learning algorithms*. University of Alberta PhD thesis.

Mahmood, A. R., Sutton, R. S. (2013). Representation Search through Generate and Test. In *AAAI Workshop: Learning Rich Representations from Low-Level Sensors*.

Mahmood, A. R., Sutton, R. S., Degris, T., Pilarski, P. M. (2012). Tuning-free step-size adaptation. *International Conference on Acoustics, Speech and Signal Processing*, pp. 2121–2124. IEEE.

McCarthy, J. 1997. What is Artificial Intelligence? Available electronically at http:// www-formal.stanford.edu/jmc/whatisai/whatisai.html.

McCloskey, M., Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation 24*:109–165. Academic Press.

Nikishin, E., Schwarzer, M., D'Oro, P., Bacon, P. L., Courville, A. (2022). The primacy bias in deep reinforcement learning. *International Conference on Machine Learning*, PMLR. pp. 16828–16847.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks 113*:4–71.

Rahman, P. (2020). *Toward Generate-and-Test Algorithms for Continual Feature Discovery*. University of Alberta MSc thesis.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision 115*(3):211-252.

Schraudolph, N. N. (1999). Local gain adaptation in stochastic gradient descent. In *Proceedings of the International Conference on Artificial Neural Networks*, pp. 569–574. IEEE, London.

Schraudolph, N. N. (2002). Fast curvature matrix-vector products for second-order gradient descent. *Neural Computation 14*(7):1723–1738.

Schweighofer, N., Arbib, M. A. (1998). A model of cerebellar metaplasticity. *Learning and Memory 4*(5):421–428.

Sutton, R. S. (1981). Adaptation of learning rate parameters. In: *Goal Seeking Components for Adaptive Intelligence: An Initial Assessment*, by A. G. Barto and R. S. Sutton. Air Force Wright Aeronautical Laboratories Technical Report AFWAL- TR-81-1070. Wright-Patterson Air Force Base, Ohio 45433.

Sutton, R. S. (1982). A theory of salience change dependent on the relationship between discrepancies on successive trials on which the stimulus is present. Unpublished report.

Sutton, R. S. (1992). Adapting bias by gradient descent: An incremental version of delta-bar-delta. *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 171–176, MIT Press.

Sutton, R. S. (2009). The grand challenge of predictive empirical abstract knowledge. In *Working Notes of the IJCAI-09 Workshop on Grand Challenges for Reasoning from Experiences*.

Sutton, R. S. (2022). A History of Meta-gradient: Gradient Methods for Meta-learning. ArXiv:2202.09701.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, Cs., Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. *International Conference on Machine Learning*.

Sutton, R. S., Bowling, M., Pilarski, P. M. (2022a). The Alberta Plan for AI Research. arXiv:2208.11173.

Sutton, R. S., Machado, M. C., Holland, G. Z., Timbers, D. S. F., Tanner, B., White, A. (2022b). Reward-Respecting Subtasks for Model-Based Reinforcement Learning. arXiv:2202.03466.

Sutton, R. S., Mahmood, A. R., White, M. (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research 17*(1):2603-2631.

Sutton, R. S., Whitehead, S. D. (1993). Online learning with random representations. *International Conference on Machine Learning,* pp. 314-321.

Thrun, S., Pratt, L. (1998). *Learning to Learn*. Springer, Boston, MA.

Van Seijen, H., Mahmood, A. R., Pilarski, P. M., Machado, M. C., Sutton, R. S. (2016). True online temporal-difference learning. *Journal of Machine Learning Research 17*(1):5057-5096.

Van Seijen, H., Sutton, R. S. (2014). True online TD(λ). *International Conference on Machine Learning* (pp. 692-700). PMLR.

Xu, Z., van Hasselt, H. P., Silver, D. (2018). Meta-gradient reinforcement learning. In *Advances in Neural Information Processing Systems* (pp. 2396–2407).

Zevin, J. D., Seidenberg, M. S. (2002). Age of acquisition effects in word reading and other tasks. *Journal of Memory and Language 47*:1–29.