Is there *Value* in Reinforcement Learning?

Lior Fox
Gatsby Computational Neuroscience Unit
University College London
lior.fox@ucl.ac.uk

Yonatan Loewenstein
Edmond and Lilly Safra Center for Brain Sciences
Hebrew University, Jerusalem
yonatan@huji.ac.il

Abstract

Action-values play a central role in popular Reinforcement Learing (RL) models of behavior. Yet, the idea that action-values are explicitly represented has been extensively debated. Critics had therefore repeatedly suggested that policy-gradient (PG) models should be favored over value-based (VB) ones, as a potential solution for this dilemma. Here we argue that this solution is unsatisfying. This is because PG methods are not, in fact, "Value-free" – while they do not rely on an explicit representation of Value for *acting* (stimulus-response mapping), they do require it for *learning*. Hence, switching to PG models is, *per se*, insufficient for eliminating Value from models of behavior. More broadly, the requirement for a representation of Value stems from the underlying assumptions regarding the optimization objective posed by the standard RL framework, not from the particular algorithm chosen to solve it. Previous studies mostly took these standard RL assumptions for granted, as part of their conceptualization or problem modeling, while debating the different methods used to optimize it (i.e., PG or VB). We propose that, instead, the focus of the debate should shift to critically evaluating the underlying modeling assumptions. Such evaluation is particularly important from an experimental perspective. Indeed, the very notion of Value must be reconsidered when standard assumptions (e.g., risk neutrality, full-observability, Markovian environment, exponential discounting) are relaxed, as is likely in natural settings. Finally, we use the Value debate as a case study to argue in favor of a more nuanced, algorithmic rather than statistical, view of what constitutes "a model" in cognitive sciences. Our analysis suggests that besides "parametric" statistical complexity, additional aspects such as computational complexity must also be taken into account when evaluating model complexity.

Keywords: policy-gradient, value-based, behavioral modelling, representations

Acknowledgements

We thank Lotem Elber-Dorozko and Ohad Dan for discussions, and for critically evaluating an earlier version of this text. This work was supported by the Gatsby Charitable Foundation.

1 Reinforcement Learning models

The Reinforcement Learning (RL) framework has been widely used to model the behavior of animals and humans in decision-making tasks (for reviews, see [8, 20, 18, 10]). The formalism conveniently fits into experimental setups: it consists of an agent-environment interaction loop, characterized by states (of the environment), actions (taken by the agent, by observing the current state and choosing an action from its policy), and rewards (associated with state-action pairs). Besides the descriptive element, RL models typically come with a normative assumption regarding the agent's goal – maximizing expected cumulative (and often temporally discounted) reward.

So-called "model-free" RL algorithms provide the powerful guarantee that even if the rules underlying state-transitions and rewards are unknown to the agent, it can still learn to act optimally, even without trying to explicitly learn aforementioned rules. These algorithms largely fall into two families, value-based (VB) and policy-gradient (PG). VB methods are based on the insight that the notion of optimality defined above admits a *Dynamic Programming* solution [3], through the well-known Bellman optimality equations. Solving the optimality equations explicitly would require an access to the state-transition and reward rules (the "model"). Instead, VB methods solve these implicitly – by online approximating the dynamic programming solution from observations. This typically relies on temporal-difference (TD) style learning driven by reward prediction error. A key structure in these algorithms is, as the name suggests, the policy's action-values. These measure, for any given state-action pair, the expected cumulative reward achieved by starting at this state-action, and then following the policy for all future choices. Therefore, by construction, a policy for which the action-values are maximal will solve the optimization problem.

PG methods, in their basic form, do not explicitly make use of the dynamic programming structure. Instead, they solve the optimization problem by calculating, from observations, a stochastic estimate of the gradient of the objective function with respect to the set of parameters defining a policy, and update the parameters in the direction of this (estimated) gradient [28, 27].

2 The arguments against Value

Despite (and, perhaps, also thanks to) their predominance, the extent to which RL algorithms are good models of natural behavior remains debated [5, 23, 6]. Within the larger debate, the notion of an explicit representation of Value has been particularly contested on several different levels. Criticism has been made of both the assumption that biological behavior is primarily guided by top-down optimality considerations [22, 25], as well as the assumption of a "common currency" that is used for (numerical) evaluation of different actions [12]. Moreover, it has been shown that the evidence for Value representation in the brain is weaker than previously realized [15, 21, 9]. Finally, the stimulus-response mapping in VB models is "indirect", relying on the hypothesized (and latent) Value variables, internal to the agent [4].

Following these criticisms, several authors have argued that PG models should be favored as the go-to modeling approach of biological behavior. The basis for these arguments is that PG methods could account for most experimental observations attributed traditionally to VB methods, while being conceptually simpler, requiring less assumptions, and (so goes the argument) eliminate the questionable Value construct from the models [18, 12, 4].

At a first glance, the suggestion is appealing. Indeed, PG methods are more "direct": they directly update the variables defining the optimization problem (the policy). They might also look appealing from the perspective that it is often (much) easier finding out *which* action is better, than by precisely *how much* it is better (i.e., evaluating numerically the advantage that one action has over another) [24, 14]. In the next section, however, we will argue that switching to a PG model doesn't answer, *per se*, the criticisms outlined before. This is simply because the switch to a PG model does *not* eliminate the requirement for a representation of Value.

Before moving on to our main argument, it should be made clear that we are not trying to argue that the criticisms made against Value are somehow irrelevant. On the contrary, we believe many of these important, and call for a critical re-assessments of RL modeling in neuroscience, as we will detail later.

3 Policy Gradient Methods are not "Value-free"

In basic forms of PG, the agent does not maintain a persistent representation of values that is used for *acting*; rather, it directly parameterizes a policy which maps states to actions. Nevertheless, a representation of Value is required for *learning*, in order to guide the updates for this policy. Intuitively, PG can be thought of as rolling-out actions according to the current policy, then updating the policy such that the probability of successful actions is increased the next time around (and conversely for unsuccessful actions). Crucially, how "successful" a particular action was is measured by its Value. Formally, this is evident in the well-known PG theorem [27]:

$$\nabla_\theta J(\pi_\theta) = \mathop{\mathbb{E}}_{s,a}\left[Q^{\pi_\theta}(s,a)\,\nabla_\theta \log \pi_\theta(s,a)\right] \tag{1}$$

Where π_θ is a policy parameterized by θ, $J(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$ is the objective function, r is the reward function, Q^{π_θ} is the (state-action) value function of the policy, and γ is the discount factor.

Different PG algorithms differ in how they estimate the Values needed for the update rule. For example, in REINFORCE [28], a simple sampling-based estimate (observed cumulative rewards) is used, resulting in the following learning-rule:

$$\nabla_\theta J(\pi_\theta) \approx \sum_t \nabla_\theta \log \pi_\theta(a_t|s_t) R_t \tag{2}$$

where R_t is the *empirical* return at time t, collected in an observed trajectory, namely $R_t = \sum_{\tau=0}^{\infty} \gamma^\tau r(s_{t+\tau}, a_{t+\tau})$. Note that the quantity R_t is, by definition, a sampling-based estimate of $Q^{\pi_\theta}(s_t, a_t)$.

Crucially, R_t cannot be replaced by $r(s_t, a_t)$ in the learning rule: such modification will, in general, lead to learning a sub-optimal policy [17]. One special case in which this replacement *is* valid is in multi-armed bandit problems. Because bandit problems are extremely common in behavioral experiments, it is not uncommon for authors to analyze and demonstrate different approaches in a bandit. But this might have contributed to a misleading interpretation that in PG methods, the immediate reward is the direct learning signal or regulator: the only reason this holds in bandit problems, is that in those problems (expected) reward and value are one and the same. One (tangential) lesson is that in order to fully appreciate the implications and predictions of RL models in experimental works, it is important to analyze them in MDPs that are richer than bandit problems alone [11].

In this sense, PG methods are not "Value-free": the agent must still represent action-values as part of its learning.[1] The main technical difference is that these action-value representations are "volatile" – they are not kept from one episode to the next, and are not being used to update or maintain some persistent representation of value that can be queried by the policy at acting time.

It is instructive to consider VB methods following the same considerations. While in these models there is no independent representation of the policy, they are not "policy free" – they simply derive a behavioral policy from the estimated values. That is, their policy component is "volatile", rather than an updated, persistent representation. The complementary aspect is that while behavior (the mapping of stimulus to response; states to actions) in VB models is indirect, compared to PG models, they can potentially have direct learning (the mapping of stimulus to change in parameters), while learning is more indirect in PG models. This is because in VB models, learning can truly be fully online – each observation from the environment (a transition from state-action to state-action along with the observed reward) instantaneously drives an update of the (estimated) value function, and the agent does not have to maintain a trace of entire trajectories in memory.

Viewed this way, PG and VB methods can both be understood as an instance of the general RL technique that Sutton and Barto termed Generalized Policy Iteration (GPI) [26]. GPI consists of two steps that are performed iteratively. One step is to measure, estimate, or compute the performance of the current policy. The second is to modify the policy in such a way as to (perhaps slightly) increase this performance. These steps are often called policy *Evaluation* and policy *Improvement*. The notions of policy evaluation and improvement are tightly related to those of *actor* and *critic* "modules" in an agent architecture. Today, the term "Actor-Critic" had become associated with a particular style of RL algorithms, explicitly combining PG and Value-based methods. Here we use these terms more broadly, to denote any component of the agent or architecture involved in improvement and evaluation (in fact, the idea of such double architecture predates both VB and PG methods, e.g. [1]; for historical review see [2]). With this interpretation, "pure" VB and PG methods can both be understood as forms of Actor-Critic, which differ in the way they implement the abstract idea of GPI. In PG the "actor" is explicitly modeled parametrically, while the "critic" is a simple, non-parametric and volatile component. In VB methods the "critic" is explicitly modeled parametrically, while the "actor" is a simple, volatile, derived mapping of the critic.

4 Are theories of Value necessary?

Within the context of the standard RL framework (maximizing expected cumulative discounted reward in a Markovian environment), the fact that Value appears in both PG and VB methods should, perhaps, come as no surprise. After all, both methods attempt to optimize the same objective, and this objective is *defined* in terms of Value. Thus, if one aims at eliminating, or at least reconsidering, this variable from models of operant learning, it is crucial to start at the foundations: the posited optimization problem, rather than at the details of the particular optimization methods. Moreover, from a theoretical perspective, it is becoming clearer that PG and VB methods are not as distinct as traditionally appreciated. Not only do they converge (in general) at the same solution of the optimal policy, but also, in some cases, aspect of their transient learning dynamics can be mapped to each other [13, 19].

[1] In some PG variants, value is not *directly* represented, and instead eligibility traces are used to track visits of state-actions throughout the trajectory [e.g. 2]. This is conceptually similar to way that a value-function can be computed from the successor representation (SR) [7]. Note that in both cases, value can be read-out directly by some mathematical transformation. These PG variants (unlike the SR) have not been popular in the behavioral RL field, hence we focus our discussion on the more common, REINFORCE-like, variants.

More fundamentally, the existence of a Value function relies on the Bellman Equations, which in turn depend on a set of non-trivial assumptions about the problem. These are often taken for granted by adopting standard RL as a conceptual framework for studying behavior. Yet, in the learning and behavior of biological agents, all of these assumptions (for example exponential discounting and risk neutrality) have been challenged before, and their violations preclude the existence of Value in the first place. If these assumptions are relaxed, significant modifications must be made in order for any algorithm to solve the modified optimization problem(s). Recent studies in RL theory have demonstrated that basic insights from both methods can be modified in this way. For example, PG methods can still work as long the agent is able to evaluate how "successful" an entire trajectory is (even if this cannot be decomposed into a sum of evaluations of individual actions) [30], or in partially observable scenarios, where alternative notions of optimality are considered [17, 16]. Similarly, VB methods can also be adapted to handle non-standard cases, provided a more general notion of Value (and even of the basic underlying "reward" generating the Value function) is being used [29].

We therefore conclude that it is instructive to shift our attention to questions about the nature and properties of Value as implied by the behavior (or neural activity) of biological agents, rather than asking whether or not Value in its classical sense is being represented. This shift might help resolve some, though not all, of the criticisms against Value representation in the brain. Even if Value is maintained in the models, going beyond the superficial PG-or-VB debate could help emphasize a new set of questions approachable within the RL framework, that had previously received less attention, but have potentially important implications for the psychological and neural basis of operant learning. These questions might include, for example, temporal-difference ("bootstrapping") versus monte-carlo methods, 1-step backups versus trajectory-based updates, and on-policy versus off-policy learning.

5 What's in a model?

We conclude by taking another look at the question of model simplicity. It is a tenet of scientific thinking that when comparing competing explanations for a phenomenon, simpler explanations should be preferred over complex ones (provided both are able to explain the phenomenon). As discussed earlier, this was one of the arguments proposed for favoring PG over VB methods [4].

And yet, there is no universal principle for determining an explanation's (or model's) complexity. A widely accepted practice in contemporary brain and behavioral sciences is to adopt a *statistical* notion of complexity, relying on the number of free parameters that have to be estimated from data. These criteria on their own ignore, however, other aspects of complexity that are just as relevant in modeling cognitive phenomena, such as *computational* complexity.

Consider again a PG and a VB algorithm for modeling behavior. From a statistical perspective, the VB model might require additional parameters (responsible for the mapping between Value, which is latent, and actions, which are observed) compared to the PG model. However, from a space-complexity perspective, VB model is simpler: the agent can learn in a fully online way, without storing entire trajectories in memory, as is required by the PG model. VB methods do not gain this simplicity for free: as previously discussed, these methods make more explicit assumptions about the problem, so they can offload parts of the complexity to these assumptions (specifically, the assumptions that enable a dynamic programming solution). These extra assumptions can be viewed as yet *another* axis of model complexity.

There might not be an "off-the-shelf" method to account for such multi-axis complexity, but acknowledging it is nonetheless important. Computational models such as RL algorithms are often interpreted as *mechanistic*, at least on the level of the cognitive architecture (and often also at the level of neural implementation). If such interpretations are to be taken seriously, then all algorithmic components of the model, including "non-parametric" ones (e.g., the sampling-based estimation of Value in REINFORCE) must be factored-in when evaluating model complexity. A PG *agent* is not just the neural network (or lookup table) policy, and a VB agent is not just the value neural network (or table): they are entire *algorithms* for updating, manipulating, and using these networks, and it is these algorithms which serve, effectively, as our model for the biological organism in the experiment. We believe that this more holistic interpretation of "a model" is needed in other domains where machine-learning algorithms are serving as quantitative models in cognitive sciences.

References

[1] A. G. Barto, R. S. Sutton, and C. W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, 1983.

[2] J. Baxter and P. L. Bartlett. Infinite-horizon policy-gradient estimation. *journal of artificial intelligence research*, 15:319–350, 2001.

[3] R. E. Bellman. *Dynamic programming*. Princeton Landmarks in Mathematics and Physics. Princeton University Press, Princeton, NJ, July 2010 (1957).

[4] D. Bennett, Y. Niv, and A. J. Langdon. Value-free reinforcement learning: policy optimization as a minimal model of operant behavior. *Current Opinion in Behavioral Sciences*, 41:114–121, 2021. Value based decision-making.

[5] O. Dan and Y. Loewenstein. From choice architecture to choice engineering. *Nature communications*, 10(1):2808, 2019.

[6] O. Dan, O. Plonsky, and Y. Loewenstein. Behavior engineering using quantitative reinforcement learning models. *Nature Communications*, 16(1):4109, 2025.

[7] P. Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

[8] P. Dayan and Y. Niv. Reinforcement learning: The good, the bad and the ugly. *Current Opinion in Neurobiology*, 18(2):185–196, 2008. Cognitive neuroscience.

[9] L. Elber-Dorozko and Y. Loewenstein. Striatal action-value neurons reconsidered. *eLife*, 7:e34248, may 2018.

[10] L. Fox, O. Dan, L. Elber-Dorozko, and Y. Loewenstein. Exploration: from machines to humans. *Current Opinion in Behavioral Sciences*, 35:104–111, 2020. Curiosity (Explore vs Exploit).

[11] L. Fox, O. Dan, and Y. Loewenstein. On the computational principles underlying human exploration. Oct. 2023.

[12] B. Y. Hayden and Y. Niv. The case against economic values in the orbitofrontal cortex (or anywhere else in the brain). *Behavioral Neuroscience*, 135(2):192, 2021.

[13] S. M. Kakade. A natural policy gradient. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.

[14] C. Laidlaw, S. Russell, and A. Dragan. Bridging RL theory and practice with the effective horizon. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[15] J. Li and N. D. Daw. Signals in human striatum are appropriate for policy update rather than value prediction. *Journal of Neuroscience*, 31(14):5504–5511, 2011.

[16] Y. Loewenstein, D. Prelec, and H. S. Seung. Operant matching as a nash equilibrium of an intertemporal game. *Neural Computation*, 21(10):2755–2773, 10 2009.

[17] Y. Loewenstein and H. S. Seung. Operant matching is a generic outcome of synaptic plasticity based on the covariance between reward and neural activity. *Proceedings of the National Academy of Sciences*, 103(41):15224–15229, 2006.

[18] G. Mongillo, H. Shteingart, and Y. Loewenstein. The misbehavior of reinforcement learning. *Proceedings of the IEEE*, 102(4):528–541, 2014.

[19] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans. Bridging the gap between value and policy based reinforcement learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[20] Y. Niv. Reinforcement learning in the brain. *Journal of Mathematical Psychology*, 53(3):139–154, 2009.

[21] J. P. O'Doherty. The problem with value. *Neuroscience & Biobehavioral Reviews*, 43:259–268, 2014.

[22] O. Plonsky and I. Erev. Learning in settings with partial feedback and the wavy recency effect of rare events. *Cognitive Psychology*, 93:18–43, 2017.

[23] M. Rosenberg, T. Zhang, P. Perona, and M. Meister. Mice in a labyrinth show rapid learning, sudden insight, and efficient exploration. *eLife*, 10:e66175, jul 2021.

[24] Ö. Şimşek, S. Algorta, and A. Kothiyal. Why most decisions are easy in tetris—and perhaps in other sequential decision problems, as well. In M. F. Balcan and K. Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1757–1765, New York, New York, USA, 20–22 Jun 2016. PMLR.

[25] G. Suri, J. J. Gross, and J. L. McClelland. Value-based decision making: An interactive activation perspective. *Psychological Review*, 127(2):153, 2020.

[26] R. S. Sutton and A. G. Barto. *Reinforcement learning : an introduction*. Second edition, 2018.

[27] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. In S. Solla, T. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12. MIT Press, 2000.

[28] R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[29] T. Zahavy, B. O' Donoghue, G. Desjardins, and S. Singh. Reward is enough for convex mdps. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 25746–25759. Curran Associates, Inc., 2021.

[30] J. Zhang, A. Koppel, A. S. Bedi, C. Szepesvari, and M. Wang. Variational policy gradient method for reinforcement learning with general utilities. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4572–4583. Curran Associates, Inc., 2020.