Toward Provably Unbiased Temporal-Difference Value Estimation

Roy Fox
Department of Computer Science
University of California, Irvine
royf@uci.edu

Abstract

Temporal-difference learning algorithms, such as Q-learning, maintain and iteratively improve an estimate of the value that an agent can expect to gain in interaction with its environment. Unfortunately, the value updates in Q-learning induce positive bias that causes it to overestimate this value. Several algorithms, such as Soft Q-learning, regularize the value updates to reduce this bias, but none provide a principled schedule of their regularizers such that early in the learning process updates are more agnostic, but then increasingly trust the value estimates more as they become more certain during learning. In this paper, we present a closed-form expression for the regularization coefficient that completely eliminates bias in entropy-regularized value updates, and illustrate this theoretical analysis using a proof-of-concept algorithm that approximates the conditions for unbiased value estimation.

1 Introduction

Reinforcement learning (RL) algorithms infer a control policy for an agent interacting with its environment from data consisting of agent–environment interaction and the reward obtained in each time step of this interaction. The control policy should select action a when the current state is s, if this leads to high expected future accumulated reward, also known as the value $Q(s, a)$. A popular way of learning a successful control policy is therefore via estimating the value function.

Temporal-difference (TD) algorithms [1] maintain and iteratively improve a value estimate, by updating the estimate in a given time step of the interaction based on the estimated value in a later time step. A key example is the Q-learning algorithm [2], which adjusts the value estimate toward the sum of the immediate reward and the estimated value of the apparently best (greedy) action in the following time step.

Most TD algorithms are consistent, up to representational concerns, in the sense that a full tabular parametrization of the value estimate is guaranteed asymptotic convergence to the optimal value. Before convergence, however, Q-learning is known to be positively biased, that is, to overestimate the value. This makes the value estimate unreliable for downstream decisions, such as the exploration policy that collects further interaction data.

Several works have addressed this bias and proposed algorithms with reduced bias [3, 4, 5, 6, 7]. In particular, Fox et al. [7] have shown that the Soft Q-learning (SQL) algorithm [8, 7, 9] is an unbiased value estimator, provided oracle scheduling of a single parameter — the inverse-temperature β — throughout the learning process. Intuitively, early in the learning process when uncertainty is high, β should be low to induce "softer", more agnostic updates. As uncertainty decreases through learning, β should be increased to induce "harder", more trusting value updates. Despite this intuition, most

Optimization Foundations for Reinforcement Learning Workshop at NeurIPS 2019, Vancouver, Canada.

publications of this type still use constant β, often simply 1 [8, 9, 10, 11, 12, 13, 14], possibly for lack of a principled recipe for scheduling β [7, 15, 16].

In this work, we take a major step toward realizing an *annealing* schedule by which the inverse-temperature β is adjusted to induce an unbiased value estimator. Our theoretical analysis applies to domains with a finite action space, once they are transformed into equivalent domains with a binary action space. We find a closed-form expression for the value of β that completely eliminates bias in entropy-regularized value updates, subject to conditions on the distribution of the current value estimator. We propose the Entropy-regularized Q-learning (EQL) algorithm, which approximately satisfies these conditions, and experiment in a small domain to illustrate the theoretical analysis.

The main purpose of this work is to highlight the often overlooked benefit of having a scheduled, rather than fixed, inverse-temperature in information-theoretic RL methods; and more generally of annealing regularizers as optimization progresses. The theoretical analysis we present is, to our knowledge, the first principled approach to provably unbiased TD value estimation, and may lead to even more practical algorithms that perform iterative updates with just the right amount of confidence.

2 Preliminaries

We consider a Markov Decision Process (MDP) with state transition distribution p, such that $p(s'|s,a)$ is the probability that the environment state becomes s' after action a is taken in state s. An agent controls the process using policy π, by observing the current state s and selecting an action a with probability $\pi(a|s)$. Each step of the agent–environment interaction is annotated with a real reward r that depends only on that step's state and action.

This work focuses on MDPs with binary action spaces, $a \in \{0, 1\}$. We note that any MDP with a finite action space can be transformed into an equivalent one with binary actions, by encoding each action as a sequence of bits and augmenting the state to maintain the prefix of this sequence before it is complete and the action applied.

Given a dataset of interaction steps (s, a, r, s'), a learning algorithm should infer a control policy that is expected to achieve high return, i.e. accumulated discounted reward $R = \sum_{t \geqslant 0} \gamma^t r_t$, where t counts time steps in an interaction process and $0 \leqslant \gamma < 1$ is a discount factor. Value-based algorithms achieve this by maintaining an estimate of the state–action value function

$$Q^\pi(s,a) = \mathbb{E}_{a_t|s_t \sim \pi \ \ \forall t>0}[R|s_0 = s, a_0 = a]$$

of some policy π. The Q-learning algorithm aims to estimate the value function of the optimal policy, $Q^* = \max_\pi Q^\pi$, by updating on each experience tuple (s, a, r, s')

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)),$$

with $\alpha > 0$ a learning rate.

An issue with this approach is that, when there's uncertainty in the value estimate, $\max_{a'} Q(s',a')$ can overestimate the true optimal value and introduce bias into the learning process. This is known as the winner's curse, and is captured by Jensen's inequality

$$\mathbb{E}[\max_a Q(s,a)] \geqslant \max_a \mathbb{E}[Q(s,a)].$$

A greedy update policy π_Q that selects the apparently best a' for $Q(s',a')$ therefore puts too much trust in the noisy estimator Q. Fox et al. [7] proposed to consider an additional cost term, which penalizes the update policy for deviating from an agnostic (e.g., uniform) prior policy π_0

$$\tilde{r}_t = r_t - \frac{1}{\beta} \log \frac{\pi_Q(a_t|s_t)}{\pi_0(a_t|s_t)},$$

with β a tradeoff coefficient, in analogy to the inverse of the temperature in statistical physics. The optimal policy is then the softmax distribution

$$\pi_Q(a|s) = \operatorname*{sm}_a(\beta Q(s,\cdot); \pi_0(\cdot|s)) \stackrel{\text{def}}{=} \frac{\pi_0(a|s)\exp(\beta Q(s,a))}{\sum_{a'} \pi_0(a'|s)\exp(\beta Q(s,a'))}.$$

We denote the softmax distribution by $\mathrm{sm} : \mathbb{R}^k \to \Delta^k$, where Δ^k is the simplex of k-dimensional probability vectors, despite the name `softmax` that some software frameworks call it, to distinguish it from the softmax expectation

$$\operatorname*{softmax}_a(Q;\beta) \stackrel{\text{def}}{=} \mathbb{E}_{a\sim \mathrm{sm}_a(\beta Q)}[Q(a)],$$

which has the same type $\mathbb{R}^k \to \mathbb{R}$ as the $\max$ operator.

Substituting the update policy π_Q into the update equation yields the Soft Q-learning (SQL) algorithm

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r + \frac{\gamma}{\beta}\log \mathbb{E}_{a'|s'\sim\pi_0}[\exp(\beta Q(s',a'))] - Q(s,a)\right).$$

In the following, we assume a uniform prior π_0, and omit it.

SQL replaces the "hard" $\max$ operator of Q-learning with a "soft" log-partition function, also known as mellowmax [17]. Similarly to $\mathrm{softmax}$, the log-partition operator has the property that, as $\beta \to 0$, it converges to the prior $\mathbb{E}_{a'|s'\sim\pi_0}[Q(s',a')]$, and as $\beta \to \infty$, it returns to the hard $\max$. Since the log-partition function is continuous in β, Fox et al. [7] concluded by the intermediate value theorem that there must be some β (generally different for each update iteration) for which the update is unbiased. This parameter should generally increase throughout the learning process, i.e. the temperature should decrease, hence the name *annealing* for such a process. However, the existence proof of an unbiased β schedule is nonconstructive, and to our knowledge a principled annealing schedule for TD algorithms has so far been unknown.

In this work, we replace Q-learning's hard $\max$ with $\mathrm{softmax}$, rather than with SQL's log-partition. This alternative is less well motivated than SQL, but allows analysis that reveals an unbiased β schedule.

3 Why Unbiased Estimation

Estimation bias is of course not an issue when the exact bias is known, because it can simply be subtracted. When we discuss biased estimators, what we really mean is that the uncertainty in the bias (due to either the model or the data) is so much larger than the uncertainty in the estimator itself, that subtracting a best guess of the bias would result in a worse estimator.

In light of this, we should be cautious of making decisions based on biased estimators. When we select the maximum of two estimates, we implicitly take the biases of their estimators into consideration, and the result may be much more uncertain than the estimators themselves.

An example of such decision making is an exploration policy interacting with the environment to collect more experience. If this policy is based on a biased value estimator, it may give undue preference to actions whose value is positively biased. We emphasize that this is different than *optimism under uncertainty*, where under-explored choices are intrinsically motivated. Bias is not the same as, for example, an upper confidence bound, in that it is not a principled and calculated preference for optimism. Instead, bias can cause too much optimism that prevents exploitation, or non-uniform optimism that starves some pathways from being well explored.

4 Unbiased Value Estimation

4.1 Single-step unbiased estimation

We start with the case of a single-step binary action $a \in \{0,1\}$, and an estimator of its value $Q(a) \sim \mathcal{N}(\mu_a, \sigma_a^2)$ having jointly Gaussian distribution. We are interested in an unbiased estimator of the maximal value $\max_a \mu_a$.

Proposition 1. *Let $a \in \{0,1\}$, and $Q(a)$ a value estimator such that $Q(0)-Q(1) \sim \mathcal{N}(\mu_A, \sigma_A^2)$. For $\beta = \frac{2|\mu_A|}{\sigma_A^2}$, the softmax expectation of Q is an unbiased estimator of the optimal value $\max_a \mathbb{E}[Q(a)]$.*

Proof. Without loss of generality, assume $\mu_0 \geqslant \mu_1$, and let the estimated advantage of action 0 over action 1 be $A = Q(0) - Q(1) \sim \mathcal{N}(\mu_A, \sigma_A^2)$, with $\mu_A \geqslant 0$. Then

$$\begin{aligned}
&\mathbb{E}_Q[\operatorname*{softmax}_a(Q;\beta)] \\
&= \mathbb{E}_Q\left[\frac{\exp(\beta Q(0))Q(0) + \exp(\beta Q(1))Q(1)}{\exp(\beta Q(0)) + \exp(\beta Q(1))}\right] \\
&= \mathbb{E}_Q\left[Q(0) - \frac{\exp(\beta Q(1))A}{\exp(\beta Q(0)) + \exp(\beta Q(1))}\right] \\
&= \mathbb{E}_Q\left[Q(0) - \frac{\exp(-\frac{1}{2}\beta A)A}{\exp(\frac{1}{2}\beta A) + \exp(-\frac{1}{2}\beta A)}\right] \\
&= \mu_0 - \int_{-\infty}^{\infty} dA \frac{1}{\sqrt{2\pi}\sigma_A} \exp\left(-\frac{(A-\mu_A)^2}{2\sigma_A^2}\right) \frac{\exp(-\frac{1}{2}\beta A)A}{\exp(\frac{1}{2}\beta A) + \exp(-\frac{1}{2}\beta A)} \\
&= \mu_0 - \int_{-\infty}^{\infty} dA \frac{1}{\sqrt{2\pi}\sigma_A} \exp\left(-\frac{A^2+\mu_A^2}{2\sigma_A^2}\right) \frac{A}{\exp(\frac{1}{2}\beta A) + \exp(-\frac{1}{2}\beta A)} \qquad (1) \\
&= \mu_0,
\end{aligned}$$

where in (1) $\beta = \frac{2\mu_A}{\sigma_A^2}$ is substituted in the numerator to cancel out the cross term $\exp(\frac{A\mu_A}{\sigma_A^2})$, and the integrand can be seen to be antisymmetric. □

This simple case already yields important intuition. As the advantage estimator A becomes more certain during the learning process, β should increase inversely to its variance. Since the variance of an estimator often decreases inversely to the amount of evidence, this result provides a formal justification for the heuristic that schedules β linearly in the number of Q updates [7, 16].

Furthermore, if we consider the update policy $\pi = \mathrm{sm}_a(\beta Q)$, we see that

$$\log\frac{\pi(0)}{\pi(1)} = \beta A = \frac{2A\mu_A}{\sigma_A^2} \sim \mathcal{N}\left(2\left(\frac{\mu_A}{\sigma_A}\right)^2, 4\left(\frac{\mu_A}{\sigma_A}\right)^2\right).$$

Thus, the log-odds of selecting the greedy action 0 also increases inversely to A's variance.

4.2 Entropy-regularized Bellman operator

We turn to present a TD algorithm for unbiased value estimation, based on an entropy-regularized Bellman operator.

Definition 2 (Regularized Bellman operator)**.** *The regularized Bellman operator $\mathcal{B}^{\mathcal{F}}$ for some objective $\mathcal{F}$ is given by*

$$\mathcal{B}^{\mathcal{F}}[Q](s,a) = \mathbb{E}[r|s,a] + \gamma\,\mathbb{E}_{s'|s,a\sim p}[\mathbb{E}_{a'|s'\sim\pi_Q^{\mathcal{F}}}[Q(s',a')]],$$

where $\pi_Q^{\mathcal{F}}$ is optimal for $\mathcal{F}$, i.e. for each state s

$$\pi_Q^{\mathcal{F}}(\cdot|s) = \operatorname*{argmax}_{\pi(\cdot|s)} \mathbb{E}_{a|s\sim\pi}[\mathcal{F}_{Q,\pi}(s,a)].$$

The Bellman operator used in Q-learning is the special case where the objective is unregularized $\mathcal{F}_{Q,\pi} = Q$, so that $\pi_Q^{\mathcal{F}}$ is the greedy policy. Here we use an entropy-regularized objective

$$\mathcal{F}_{Q,\pi}(s,a) = Q(s,a) - \beta^{-1}\log\pi(a|s),$$

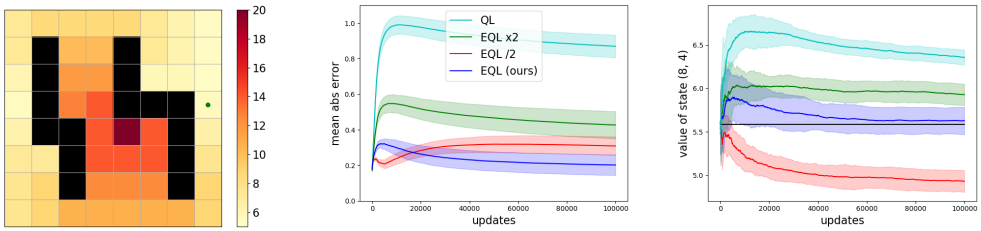
which is optimized by the softmax policy $\pi_Q(\cdot|s) = \mathrm{sm}_a(\beta Q(s,\cdot))$.

Corollary 3. *In a binary-action domain, let Q be a Gaussian-distributed unbiased estimator of Q^*, and $\beta(s)$ as in Proposition 1. Then $\mathcal{B}^{\mathcal{F}}[Q]$ for $\mathcal{F}$ the entropy-regularized objective is also an unbiased estimator of Q^*.*

Proof.

$$\begin{aligned}
\mathbb{E}_Q[\mathcal{B}^{\mathcal{F}}[Q](s,a)] &= \mathbb{E}[r|s,a] + \gamma\,\mathbb{E}_{s'|s,a\sim p}[\max_{a'} Q^*(s',a')] \qquad (2) \\
&= Q^*(s,a), \qquad (3)
\end{aligned}$$

where (2) follows from Proposition 1 and (3) from Bellman optimality. □



(a) (b) (c)

Figure 1: (a) Optimal state value in the grid world domain. Walls (black) are impassable. The absorbing goal state has the highest value (darkest red). (b) Absolute error between the mean estimate and the optimal value, averaged over all states and 10 runs. Standard deviation (shaded) is over the runs. EQL (blue) is compared to β scheduling that is twice (green) and half (red) its theoretically-motivated value, and to Q-learning having $\beta = \infty$ (cyan). (c) Bias in the state marked by a green dot in (a), averaged over the 20 estimates and 10 runs. Standard deviation (shaded) is over the estimates. The same four schedules are compared to the optimal value (black).

4.3 Entropy-regularized Q-learning

Entropy-regularized Q-learning (EQL) is an off-policy algorithm which, on experience (s, a, r, s'), updates

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \operatorname*{softmax}_{a'}(Q(s',a');\beta) - Q(s,a)),$$

where α is the learning rate and β the inverse-temperature. EQL is different than Q-learning and SQL in that the $\mathrm{softmax}$ operator is used instead of, respectively, $\max$ and the log-partition. With careful scheduling of α and β, which we detail below, EQL has the potential to maintain an approximately unbiased estimator of Q^*.

The conditions of Corollary 3 require Q to always be a Gaussian-distributed unbiased estimator of Q^*. Initializing such Q requires prior knowledge of Q^*. A reasonable assumption is that we know a range in which Q^* falls. In this case, we can initialize Q from a Gaussian distribution with standard deviation on an order larger than that range, such that the initial bias is indiscernible in the noise.

Corollary 3 also requires that we know the distribution of $A(s) = Q(s,0) - Q(s,1)$ for each state s. It is unreasonable to know this exactly for an unbiased estimator, because then we would already have the optimal policy, which selects action 0 whenever $\mu_A(s) \geqslant 0$. Instead, we estimate this distribution and use the estimate to select an approximate β. A remaining open question is how sensitive Corollary 3 is to such approximation.

A naive implementation of EQL explicitly represents the distribution of Q, for example by representing its first and second moments, or as a set of particles sampled from that distribution. For each experience step (s, a, r, s'), $\beta = \frac{2|\mu_A(s')|}{\sigma_A^2(s')}$ is recovered from Q's distribution, an estimate $Q(s', \cdot)$ is sampled, and $r + \mathrm{softmax}_{a'}(Q(s', a'); \beta)$ computed. Each update step then uses the average of sufficiently many experience steps, such that this average is approximately Gaussian-distributed due to the central limit theorem (CLT).

In summary, the update steps in EQL approximately preserve the conditions of Corollary 3, but may in practice violate them in the following ways: (1) If the estimator is biased before the update, this bias carries over to the updated estimator, but discounted by γ; (2) The updated estimator may not really have a Gaussian distribution, e.g. due to averaging too few estimates for CLT; (3) The estimation of β may not result in the prescribed value, e.g. under carried bias or misestimated uncertainty.

5 Experiments

In order to illustrate the theoretical results, we present a proof-of-concept implementation of EQL in the same small grid-world domain as [7]. In the original domain, the states are the cells in Figure 1a which are not walls (shown in black), and the actions are to move in the 8 directions. The binary-action equivalent domain has the actions spell out the binary index of the desired direction in

3 consecutive time steps, and each state is copied $1 + 2 + 4$ times to also represent the index prefix of length 0, 1, or 2. When the third bit is selected, the agent moves in the desired direction, but may end up in another cell adjacent to the desired: the agent has probabilities 0.15 to slip in each cardinal direction; 0.05 to slip in each intercardinal direction; and 0.2 to not slip and reach the desired cell. A wall stops the agent from moving or slipping in that direction.

The expected reward is 0, except in the absorbing goal state (darkest red in Figure 1a) where it is 1. The actual reward has Gaussian distribution with variance 1. The optimal expected return in each cell is shown in Figure 1a. Due to the discount factor $\gamma = 0.95$, the effective expected return in the goal state is 20.

In order to compare different update parameters, and eliminate any indirect effect this may have through the exploration policy, we use the same standard data sampling scheme [18, 7] in all experiments: we sample a state s and an action a uniformly, and then sample the reward r and the following state s' using the domain's distributions.

We initialize 20 value estimates by adding Gaussian noise with variance 1 to the optimal value function. We then schedule $\alpha = n(s,a)^{-\omega}$, where $n(s,a)$ is the number of updates of $Q(s,a)$ so far, and $\omega = 0.8$ as suggested by Even-Dar and Mansour [19]. We schedule β using the empirical mean and variance of our set of value estimates. In each update iteration, we update each of the 20 estimates once using one experience point (s, a, r, s'). We note that this scheme is not proposed as an efficient implementation of EQL, but rather only to illustrate its properties.

Figure 1b shows the absolute error between the empirical mean estimate and the optimal value, averaged over all states and 10 runs of the algorithm (the shaded area is the standard deviation over the runs). The blue curve has β scheduled according to the theory presented in Section 4. The green and red curves have β twice and half of that, respectively. The cyan curve uses the mean of 20 Q-learning estimates in each of the 10 runs, which is equivalent to setting $\beta = \infty$. These results suggest that the theoretically-motivated annealing schedule may help EQL converge faster.

Finally, Figure 1c shows the bias of the EQL estimator in a specific state (the green dot in Figure 1a) and the greedy action. The bias is averaged over the 20 estimates and 10 runs (the shaded area is the standard deviation over the estimates), and shown relative to the optimal value (black line). Note how early bias in EQL (blue curve) is later reduced: this pattern is consistent with having no new bias introduced by later updates, and early bias decay exponentially as it is discounted in each update. These results suggest that EQL may indeed reduce bias in value estimators.

6 Conclusion

In this work, we presented a closed-form expression for the inverse-temperature parameter β that makes softmax estimators unbiased for the optimal value. This β is given in terms of quantities that can be estimated empirically, namely the mean and variance of the estimator of the advantage of one action over the other in a binary-action setting. As a proof-of-concept, we developed a simple temporal-difference algorithm that maintains an approximately unbiased value estimator, and illustrated its properties in a simple domain.

Our theoretical analysis carries over completely to value function approximations that generalize better and can learn faster in large state spaces, such as neural networks [20]. While the naive implementation of EQL relied on a full tabular representation, recent advances in Bayesian deep learning provide methods for efficiently representing and training distributions over models [21, 22, 23, 24]. These methods provide powerful ways to maintain the distribution of Q needed in EQL, and will be explored in future work.

Related to our approach, Distributional RL defines a Bellman operator that operates on value distributions [25, 26]. While Distributional RL models *aleatoric* value uncertainty, i.e. resulting from the stochasticity of the process, this work is concerned with *epistemic* uncertainty, i.e. resulting from incomplete learning. An interesting open question is how to combine the two approaches.

Acknowledgements

The author thanks Noga Zaslavsky, Naftali Tishby, and Ari Pakman, for insightful discussions.

References

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

[2] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[3] Hado V Hasselt. Double q-learning. In *Advances in Neural Information Processing Systems*, pages 2613–2621, 2010.

[4] Mohammad Ghavamzadeh, Hilbert J Kappen, Mohammad G Azar, and Rémi Munos. Speedy q-learning. In *Advances in neural information processing systems*, pages 2411–2419, 2011.

[5] Donghun Lee and Warren B Powell. An intelligent battery controller using bias-corrected q-learning. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[6] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

[7] Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. In *32nd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.

[8] Michael Bloem and Nicholas Bambos. Infinite time horizon maximum causal entropy inverse reinforcement learning. In *53rd IEEE Conference on Decision and Control*, pages 4911–4916. IEEE, 2014.

[9] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1352–1361. JMLR. org, 2017.

[10] Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2775–2785, 2017.

[11] John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

[12] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[13] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

[14] Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: Imitation learning via regularized behavioral cloning. *arXiv preprint arXiv:1905.11108*, 2019.

[15] Felix Leibfried, Jordi Grau-Moya, and Haitham Bou-Ammar. An information-theoretic optimality principle for deep reinforcement learning. *arXiv preprint arXiv:1708.01867*, 2017.

[16] Jordi Grau-Moya, Felix Leibfried, and Peter Vrancx. Soft q-learning with mutual-information regularization. 2018.

[17] Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 243–252. JMLR. org, 2017.

[18] Michael J Kearns and Satinder P Singh. Finite-sample convergence rates for q-learning and indirect algorithms. In *Advances in neural information processing systems*, pages 996–1002, 1999.

[19] Eyal Even-Dar and Yishay Mansour. Learning rates for q-learning. *Journal of machine learning Research*, 5(Dec):1–25, 2003.

[20] 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, 2015.

[21] Hao Wang and Dit-Yan Yeung. Towards bayesian deep learning: A framework and some existing methods. *IEEE Transactions on Knowledge and Data Engineering*, 28(12):3395–3408, 2016.

[22] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584, 2017.

[23] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[24] Tim GJ Rudner, Vincent Fortuin, Yee Whye Teh, and Yarin Gal. On the connection between neural processes and gaussian processes with deep kernels. In *Workshop on Bayesian Deep Learning, NeurIPS*, 2018.

[25] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 449–458. JMLR. org, 2017.

[26] Clare Lyle, Marc G Bellemare, and Pablo Samuel Castro. A comparative analysis of expected and distributional reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4504–4511, 2019.