An Information-Theoretic Optimality Principle for Deep Reinforcement Learning

Felix Leibfried [1] **Jordi Grau-Moya** [1] **Haitham Bou-Ammar** [1]

Abstract

We methodologically address the problem of Q-value overestimation in deep reinforcement learning to handle high-dimensional state spaces efficiently. By adapting concepts from information theory, we introduce an intrinsic penalty signal encouraging reduced Q-value estimates. The resultant algorithm encompasses a wide range of learning outcomes containing deep Q-networks as a special case. Different learning outcomes can be demonstrated by tuning a Lagrange multiplier accordingly. We furthermore propose a novel scheduling scheme for this Lagrange multiplier to ensure efficient and robust learning. In experiments on Atari games, our algorithm outperforms other algorithms (e.g. deep and double deep Q-networks) in terms of both game-play performance and sample complexity.

1. Introduction

Reinforcement learning (Sutton & Barto, 1998) (RL) is a discipline of artificial intelligence seeking to find optimal behavioral policies that enable agents to collect maximal reward while interacting with the environment. A popular RL algorithm is Q-learning (Watkins, 1989) that operates by estimating expected cumulative rewards (Q-values). Although successful in numerous applications (Busoniu et al., 2010), standard Q-learning suffers from two drawbacks. First, due to its tabular nature in representing Q-values, it is not readily applicable to high-dimensional environments with large state and/or action spaces. Second, it initially overestimates Q-values, introducing a bias at early stages of training (Fox et al., 2016). This bias has to be "unlearned" as training proceeds, thus decreasing sample efficiency.

To address the first problem, Q-learning has been extended to high-dimensional environments by using parametric function approximators instead of Q-tables (Busoniu et al., 2010).

[1]PROWLER.io, Cambridge, UK. Correspondence to: Felix Leibfried <felix@prowler.io>.

One particularly appealing class of approximators are deep neural networks that learn "complex" relationships between high-dimensional inputs (e.g. images) and low-level actions. Building on this idea, deep Q-networks (DQNs) (Mnih et al., 2015) were proposed, attaining state-of-the-art results in large-scale domains, e.g. the Arcade Learning Environment for Atari games (Bellemare et al., 2013). Though successful, DQNs fail to address the overestimation problem, and are therefore rather sample-inefficient (van Hasselt et al., 2016).

One way of addressing Q-value overestimation is to introduce an intrinsic penalty signal in addition to instantaneous rewards. The intrinsic penalty affects the learned Q-values, eventually leading to lower estimates. Information theory provides a principled method to formalize such a penalty by interpreting the agent as an information-theoretic channel with limited transmission rate (Sims, 2010; Ortega & Braun, 2013). Specifically, the state of the environment can be interpreted as channel input, the action as channel output and the agent's reward as quality of information transmission (Genewein et al., 2015). Interestingly, in the RL setting, limits in transmission rate reflect limits in "information resources" the agent can spend to deviate from a given reference policy. The instantaneous deviation between the agent's current policy and such a reference policy directly results in an intrinsic penalty to be subtracted from the reward signal. Information-theoretic RL approaches (Azar et al., 2012; Rawlik et al., 2012; Fox et al., 2016) have already been designed for the tabular setting but do not readily apply to high-dimensional environments that require parametric function approximators.

Since we are interested in improving sample complexity of RL in high-dimensional state spaces, we contribute by adapting information-theoretic concepts to phrase a novel optimization objective for learning Q-values with deep parametric function approximators. The resultant algorithm encompasses a wide range of learning outcomes that can be demonstrated by tuning a Lagrange multiplier. We show that DQNs arise as a special case of our proposed approach. We further contribute by introducing a dynamic scheduling scheme for adapting the magnitude of intrinsic penalization based on temporal Bellman error evolution. This ultimately allows us to outperform DQN and other meth-

ods, such as double DQN (van Hasselt et al., 2016) and soft Q-learning (Schulman et al., 2017), by large margins in terms of both game score and sample complexity in the Atari domain. At the same time, our approach leads to decreased Q-value estimates, confirming our hypothesis that overestimation leads to poor performance in practice. Finally, we show further performance increase by adopting the dueling architecture from (Wang et al., 2016). In short, the contributions of this paper are:

1. applying information-theoretic concepts to high-dimensional state spaces with function approximators;
2. proposing a novel information-theoretically inspired optimization objective for deep RL;
3. demonstrating a wide range of learning outcomes for deep RL including DQNs as a special case; and
4. outperforming DQN, double DQN, and soft Q-learning in the Atari domain.

2. Reinforcement Learning

In RL, an agent, being in a state $\boldsymbol{s} \in \mathcal{S}$, chooses an action $\boldsymbol{a} \in \mathcal{A}$ sampled from a behavioral policy $\boldsymbol{a} \sim \pi_{\text{behave}}(\boldsymbol{a}|\boldsymbol{s})$, where $\pi_{\text{behave}} : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Resulting from this choice is a transition to a successor state $\boldsymbol{s}' \sim \mathcal{P}(\boldsymbol{s}'|\boldsymbol{s}, \boldsymbol{a})$, where $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the unknown state transition model, and a reward $r = \mathcal{R}(\boldsymbol{s}, \boldsymbol{a})$ that quantifies instantaneous performance. After subsequent interactions with the environment, the goal of the agent is to optimize for $\pi^{\star}_{\text{behave}}$ that maximizes the expected cumulative return $\mathbb{E}_{\pi_{\text{behave}},\mathcal{P}}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$, with t denoting time and $\gamma \in (0, 1)$ the discount factor.

Clearly, to learn an optimal behavioral policy, the agent has to reason about long term consequences of instantaneous actions. Q-learning, a famous RL algorithm, estimates these effects using state-action value pairs (Q-values) to quantify the performance of the policy. In Q-learning, updates are conducted online after each interaction $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s'})$ with the environment using

$$Q(\boldsymbol{s}, \boldsymbol{a}) \leftarrow Q(\boldsymbol{s}, \boldsymbol{a}) + \alpha \left(r + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{s}', \boldsymbol{a}') - Q(\boldsymbol{s}, \boldsymbol{a}) \right), \tag{1}$$

with $\alpha > 0$ being a learning rate. Intuitively, Equation (1) assumes an old value, i.e. the prediction $Q(\boldsymbol{s}, \boldsymbol{a})$, and corrects for its estimate based on new information, i.e. using the target $r + \gamma \max_{\boldsymbol{a}'} Q(\boldsymbol{s}', \boldsymbol{a}')$.

Optimistic Overestimation: Upon careful investigation of Equation (1), one comes to recognize that Q-learning updates introduce a bias to the learning process caused by an overestimation of the optimal cumulative rewards (van Hasselt, 2010; Azar et al., 2011; Lee & Powell, 2012; Bellemare et al., 2016; Fox et al., 2016). Specifically, the usage of the maximum operator assumes that current guesses for Q-values reflect optimal cumulative rewards. Of course, this assumption is violated, especially early in the learning process, when a relatively small number of updates has been performed. Due to the correlative effect of "bad" estimations between different state-action pairs, these mistakes tend to propagate rapidly through the Q-table and have to be unlearned in the course of further training. Though such an optimistic bias is eventually unlearned, the convergence speed (in terms of environmental interactions, i.e. sample complexity) of Q-learning is highly dependent on the quality of the initial Q-values.

The problem of optimistic overestimation only gets worse in high-dimensional state spaces, such as images in the Arcade Learning Environment. As mentioned earlier, high-dimensional representations are handled by generalizing tabular Q-learning to use parametric function approximators, e.g. deep neural networks (Mnih et al., 2015). Learning then commences by fitting weights of the approximators using stochastic gradients to minimize

$$\mathbb{E}_{\boldsymbol{s},\boldsymbol{a},r,\boldsymbol{s}'} \left[\left(r + \gamma \max_{\boldsymbol{a}'} Q_{\boldsymbol{\theta}^-}(\boldsymbol{s}', \boldsymbol{a}') - Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) \right)^2 \right]. \tag{2}$$

Here, the expectation $\mathbb{E}$ refers to samples drawn from a replay memory storing state transitions (Lin, 1993), and $Q_{\boldsymbol{\theta}^-}(\boldsymbol{s}', \boldsymbol{a}')$ denotes a DQN at an earlier stage of training. Intuitively, the minimization objective in Equation (2) resembles similarities to that used in the tabular setting. Again, old value estimates are updated based on new information, while introducing the $\max$-operator bias. Although DQNs generalize well over a wide range of input states, they are "unaware" of the aforementioned overestimation problem (Thrun & Schwartz, 1993). However, when compared with the tabular setting, this problem is even more severe due to the lack of any convergence guarantees to optimal Q-values when using parametric approximators, and the inability to explore the whole state-action space. Hence, the number of environmental interactions needed to unlearn the optimistic bias can become prohibitively expensive.

3. Addressing Optimistic Overestimation

A potential solution to optimistic overestimation in Q-learning is to add an intrinsic penalty to instantaneous rewards, thus reducing Q-value estimates. A principled way to introduce such a penalty is provided by the framework of information theory for decision-making. The rationale is to interpret the agent as an information-theoretic channel with limited transmission rate (Sims, 2010; Tishby & Polani, 2011; Ortega & Braun, 2013; Genewein et al., 2015). The environmental state $\boldsymbol{s}$ is considered as channel input,

the agent's action $\boldsymbol{a}$ as channel output and the quality of information transmission is expressed by some reward or utility function $U(\boldsymbol{s}, \boldsymbol{a})$. According to Shannon's noisy-channel coding theorem (Shannon, 1948), the transmission rate is upper-bounded by the average Kullback-Leibler (KL) divergence between the behavioral policy π_{behave} and any arbitrary reference policy with support in $\mathcal{A}$ (Csiszar & Tusnady, 1984; Tishby et al., 1999). In the following, the reference policy is denoted as prior policy π_{prior}. The KL-divergence, therefore, plays the role of a limited resource and may not exceed a maximum $K > 0$, such that $\text{KL}\left(\pi_{\text{behave}} || \pi_{\text{prior}}\right) \leq K$.

The intuition behind the information-theoretic viewpoint is that the channel aims to map input $\boldsymbol{s}$ to output $\boldsymbol{a}$, measuring the quality of the mapping in terms of $U(\boldsymbol{s}, \boldsymbol{a})$. Since the transmission rate is limited, the agent has to discard information in $\boldsymbol{s}$ that has little impact on U to obtain a utility-maximizing $\boldsymbol{a}$ without exceeding the transmission limit K. Importantly, the constraint in transmission rate directly translates into an instantaneous penalty signal leading to reduced utility, as outlined next for a one-step decision-making problem.

In a one-step scenario, we obtain the following

$$\max_{\pi_{\text{behave}}} \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_{\text{behave}}(\boldsymbol{a}|\boldsymbol{s}) U(\boldsymbol{s}, \boldsymbol{a}) \;\; \text{s.t.} \;\; \text{KL}\left(\pi_{\text{behave}} || \pi_{\text{prior}}\right) \leq K,$$

with

$$\text{KL}\left(\pi_{\text{behave}} || \pi_{\text{prior}}\right) = \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_{\text{behave}}(\boldsymbol{a}|\boldsymbol{s}) \log \frac{\pi_{\text{behave}}(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s})},$$

where the term $\log \frac{\pi_{\text{behave}}(\boldsymbol{a}|\boldsymbol{s})}{\pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s})}$ reflects instantaneous penalty[1]. The above constrained optimization problem can be expressed as a concave unconstrained objective by introducing a Lagrange multiplier $\lambda > 0$:

$$\begin{aligned} \mathcal{L}\left(\boldsymbol{s}, \pi_{\text{prior}}, \lambda\right) = \max_{\pi_{\text{behave}}} \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_{\text{behave}}(\boldsymbol{a}|\boldsymbol{s}) U(\boldsymbol{s}, \boldsymbol{a}) \\ - \frac{1}{\lambda} \text{KL}\left(\pi_{\text{behave}} || \pi_{\text{prior}}\right), \end{aligned} \tag{3}$$

where λ represents a trade-off between utility and closeness to prior information. The optimal solution can be computed in closed form as

$$\pi^{\star}_{\text{behave}}(\boldsymbol{a}|\boldsymbol{s}) = \frac{\pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s}) \exp\left(\lambda U(\boldsymbol{s}, \boldsymbol{a})\right)}{\sum_{\boldsymbol{a}'} \pi_{\text{prior}}(\boldsymbol{a}'|\boldsymbol{s}) \exp\left(\lambda U(\boldsymbol{s}, \boldsymbol{a}')\right)}. \tag{4}$$

Note that we are not the first to propose such information-theoretic principles within the context of reinforcement learning (and planning), where the utility function is usually assumed to be the expected cumulative reward, i.e. $U(\boldsymbol{s}, \boldsymbol{a}) = Q(\boldsymbol{s}, \boldsymbol{a})$. In fact, similar principles have recently received increased attention within policy search and identification of optimal cumulative reward values, as outlined in the next two paragraphs.

In policy search, information-theoretic principles similar to Equation (3) can be categorized into three classes depending on the choice of the prior policy $\pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s})$. The first class adopts a fixed prior that remains unchanged during the learning process. Entropy regularisation (Williams & Peng, 1991; Mnih et al., 2016), for example, is a special case within this class (assuming a uniform prior policy). The second class makes use of a marginal prior policy obtained by averaging the behavioral policy over all environmental states. The information-theoretic intuition, here, is to encourage the agent to neglect reward-irrelevant information in the environment (Leibfried & Braun, 2015; 2016; Peng et al., 2017). The third class assumes an adaptive prior (e.g. a policy learned at an earlier stage of training) to ensure incremental improvement steps in on-policy settings as learning proceeds (Bagnell & Schneider, 2003; Peters & Schaal, 2008; Peters et al., 2010; Schulman et al., 2015).

In optimal cumulative reward value identification, the KL-penalty is directly incorporated into Q-value estimates rather than using it for regularization. There are two distinct categories for value identification that utilize KL-constraints in different ways. The first category considers a restricted class of Markov Decision processes (MDPs), where instantaneous rewards incorporate a KL-penalty that explicitly discourages deviations from uncontrolled environmental dynamics. Such restricted MDPs enable efficient optimal value computation as outlined in (Todorov, 2009; Kappen et al., 2012). The second category comprises MDPs with intrinsic penalty signals similar to Equation (3) where deviations from a prior policy are penalized. Optimal values are either computed with generalized value iteration schemes (Tishby & Polani, 2011; Rubin et al., 2012; Grau-Moya et al., 2016), or in an RL setting similar to Q-learning (Azar et al., 2012; Rawlik et al., 2012; Fox et al., 2016).

Closest to our work are the recent approaches in (Haarnoja et al., 2017a;b; Schulman et al., 2017). It is worth mentioning that apart from the discrete action and high-dimensional state space setting, we tackle two additional problems not addressed previously. First, we consider *dynamic* adaptation for trading off rewards versus intrinsic penalties as opposed to the static scheme presented in (Haarnoja et al., 2017a;b; Schulman et al., 2017). Second, we deploy a robust computational approach that incorporates value-based advantages to ensure bounded exponentiation terms. Our approach also fits into the work of how utilising entropy for reinforcement learning connects policy search to optimal cu-

[1] Note that although we use a state-independent prior in this work, the theoretical framework for Q-value reduction remains valid for state-conditioned $\pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s})$.

mulative reward value identification (Haarnoja et al., 2017a; Nachum et al., 2017; O'Donoghue et al., 2017; Schulman et al., 2017). In this paper, however, we focus on deep value-based approaches, which show improved performance, as demonstrated in the experiments.

Due to the intrinsic penalty signal, information-theoretic Q-learning algorithms provide a principled way of reducing Q-value estimates and are hence suited for addressing the overestimation problem outlined earlier. Although successful in the tabular setting, these algorithms are not readily applicable to high-dimensional environments that require parametric function approximators. In the next section, we adapt information-theoretic concepts to high-dimensional state spaces with function approximators and demonstrate that other deep learning techniques (e.g. DQNs) emerge as a special case.

3.1. Addressing Overestimation in Deep RL

We aim to reduce optimistic overestimation in deep RL methodologically by leveraging ideas from information-theory. Since Q-value overestimations are a source of sample-inefficiency, we improve large-scale reinforcement learning where current techniques exhibit high sample complexity (Mnih et al., 2015).

To do so, we introduce an intrinsic penalty signal in line with the methodology put forward in the previous section. Before commencing, however, it can be interesting to gather more insights into the range of possible learners while tuning such a penalty. Plugging the optimal behavior policy $\pi^{\star}_{\text{behave}}$ from Equation (4) back in Equation (3) yields

$$\mathcal{L}^{\star}(\boldsymbol{s}, \pi_{\text{prior}}, \lambda) = \frac{1}{\lambda} \log \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s}) \exp\left(\lambda U(\boldsymbol{s}, \boldsymbol{a})\right).$$

The Lagrange multiplier λ steers the magnitude of the penalty signal and thus leads to different learning outcomes. If λ is large, little penalization from the prior policy is introduced. As such, one would expect a learning outcome that mostly considers maximizing utility. This is confirmed in the limit as $\lambda \to \infty$, where

$$\lim_{\lambda \to \infty} \mathcal{L}^{\star}(\boldsymbol{s}, \pi_{\text{prior}}, \lambda) = \max_{\boldsymbol{a} \in \mathcal{A}} U(\boldsymbol{s}, \boldsymbol{a}).$$

On the other hand, for small λ values, the deviation penalty is significant and the prior policy should dominate. This is again confirmed when $\lambda \to 0$, where we recover the expected utility under π_{prior}:

$$\begin{aligned}\lim_{\lambda \to 0} \mathcal{L}^{\star}(\boldsymbol{s}, \pi_{\text{prior}}, \lambda) &= \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s}) U(\boldsymbol{s}, \boldsymbol{a}) \\ &= \mathbb{E}_{\pi_{\text{prior}}}\left[U(\boldsymbol{s}, \boldsymbol{a})\right].\end{aligned}$$

Carrying this idea to deep RL by setting $U(\boldsymbol{s}, \boldsymbol{a}) = Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$, where $Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$ represents a deep Q-network, we notice that incorporating a penalty signal in the context of large-scale Q-learning with parameterized function approximators leads to

$$\mathcal{L}^{\star}_{\boldsymbol{\theta}}(\boldsymbol{s}, \pi_{\text{prior}}, \lambda) = \frac{1}{\lambda} \log \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s}) \exp\left(\lambda Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})\right).$$

We then use this operator to phrase an information-theoretic optimization objective for deep Q-learning:

$$\mathcal{J}_{\lambda}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}'}\left[\left(r + \gamma \mathcal{L}^{\star}_{\boldsymbol{\theta}^{-}}(\boldsymbol{s}', \pi_{\text{prior}}, \lambda) - Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})\right)^2\right], \tag{5}$$

where $\mathbb{E}_{\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}'}$ refers to samples drawn from a replay memory in each iteration of training, and $\boldsymbol{\theta}^{-}$ to the parameter values at an earlier stage of learning.

The above objective leads to a wide variety of learners and can be considered a generalization of current methods, including deep Q-networks (Mnih et al., 2015). Namely, if $\lambda \to \infty$, we recover the approach in (Mnih et al., 2015) that poses the problem of optimistic overestimation:

$$\mathcal{J}_{\lambda \to \infty}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}'}\left[\left(r + \gamma \max_{\boldsymbol{a}' \in \mathcal{A}} Q_{\boldsymbol{\theta}^{-}}(\boldsymbol{s}', \boldsymbol{a}') - Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})\right)^2\right].$$

On the contrary, if $\lambda \to 0$, we obtain the following

$$\mathcal{J}_{\boldsymbol{\lambda} \to 0}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}'}\left[\left(r + \gamma \sum_{\boldsymbol{a}' \in \mathcal{A}} \pi_{\text{prior}}(\boldsymbol{a}'|\boldsymbol{s}') Q_{\boldsymbol{\theta}^{-}}(\boldsymbol{s}', \boldsymbol{a}') - Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})\right)^2\right]. \tag{6}$$

Effectively, Equation (6) estimates future cumulative rewards using the prior policy as can be seen in the term $\sum_{\boldsymbol{a}' \in \mathcal{A}} \pi_{\text{prior}}(\boldsymbol{a}'|\boldsymbol{s}') Q_{\boldsymbol{\theta}^{-}}(\boldsymbol{s}', \boldsymbol{a}')$. From the above two special cases, we recognize that our formulation allows for a variety of learners, where λ steers outcomes between the above two limiting cases. Note, however, setting low values for λ introduces instead a pessimistic bias as outlined for the tabular setting in (Fox et al., 2016).

Since low λ-values introduce a pessimistic bias and large λ-values an optimistic bias, there must be a λ-value in between encouraging unbiased estimates. Unfortunately, it is not possible to compute such a λ in closed form, which is why we propose a dynamical scheduling scheme based on temporal Bellman error evolution in the next section. Note that we assume a fixed prior π_{prior} and we aim at scheduling λ. Another possibility would be to fix λ and schedule the prior action probabilities instead. The latter is however practically less convenient compared to scheduling a scalar.

4. Dynamic & Robust Deep RL

A fixed hyperparameter λ is undesirable in the course of training as the effect of the intrinsic penalty remains unchanged. Since overestimations are more severe at the start of the learning process, a dynamic scheduling scheme for λ with small values at the beginning (incurring strong penalization) and larger values towards the end (leading to less penalization) is preferable.

Adaptive λ: A suitable candidate for dynamically adapting λ in the course of training is the average squared loss (over replay memory samples) between target values $t = r + \gamma \mathcal{L}^{\star}_{\boldsymbol{\theta}^-}(\boldsymbol{s}', \pi_{\text{prior}}, \lambda)$ and predicted values $p = Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$:

$$\mathcal{J}_{\text{squared}}(t, p) = (t - p)^2$$

The rationale, here, is that λ should be inversely proportional to the average squared loss. If $\mathcal{J}_{\text{squared}}(t, p)$ is high on average, as is the case during early episodes of training, low values of λ are favored. However, if $\mathcal{J}_{\text{squared}}(t, p)$ is low on average later in training, then high λ values are more suitable for the learning process.

We therefore propose to adapt λ with a running average over the loss between targets and predictions. The running average $\mathcal{J}_{\text{avg}}$ should emphasize recent history as opposed to samples that lie further in the past since the parameters $\boldsymbol{\theta}$ of the Q-value approximator change over time. This is achieved with an exponential window and the online update

$$\mathcal{J}_{\text{avg}} \leftarrow \left(1 - \frac{1}{\tau}\right) \mathcal{J}_{\text{avg}} + \frac{1}{\tau} \mathbb{E}_{t,p} \left[\mathcal{J}_{\text{squared}}(t, p)\right], \qquad (7)$$

where τ is a time constant referring to the window size of the running average, and $\mathbb{E}_{t,p}\left[\mathcal{J}_{\text{squared}}(t, p)\right]$ is a shorthand notation for Equation (5). This running average allows one to dynamically assign $\lambda = \frac{1}{\mathcal{J}_{\text{avg}}}$ at each training iteration.

The squared loss $\mathcal{J}_{\text{squared}}(t, p)$ has an impeding impact on the stability of deep Q-learning, where the parametric approximator is a deep neural network and parameters are updated with gradients and backpropagation. To prevent loss values from growing too large, the squared loss is replaced with an absolute loss if $|t - p| > 1$ (Mnih et al., 2015). In this work, we follow a similar approach but use the Huber loss $\mathcal{J}_{\text{Huber}}(t, p)$ instead:

$$\mathcal{J}_{\text{Huber}}(t, p) = \begin{cases} \frac{1}{2}(t - p)^2 & \text{if } |t - p| < 1, \\ |t - p| - \frac{1}{2} & \text{otherwise.} \end{cases}$$

The Huber loss leads to a more robust adaptation of λ, as it uses an absolute loss for large error values instead of a squared one. Furthermore, the squared loss is more sensitive to outliers and might penalize the learning process in an unreasonable fashion in the presence of sparse but large error values.

Robust Free Energy Values: The dynamic adaptation of λ encourages learning of unbiased estimates of the optimal cumulative reward values. Presupposing $Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$ is bounded, $\mathcal{L}^{\star}_{\boldsymbol{\theta}}(\boldsymbol{s}, \pi_{\text{prior}}, \lambda)$ is also bounded in the limits of λ:

$$\mathbb{E}_{\pi_{\text{prior}}}\left[Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})\right] \leq \mathcal{L}^{\star}_{\boldsymbol{\theta}}(\boldsymbol{s}, \pi_{\text{prior}}, \lambda) \leq \max_{\boldsymbol{a} \in \mathcal{A}} Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}).$$

In practice, however, this operator is prone to computational instability for large λ due to the exponential term $\exp\left(\lambda Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})\right)$. We address this problem by amending the term $\frac{\exp(\lambda V_{\boldsymbol{\theta}}(\boldsymbol{s}))}{\exp(\lambda V_{\boldsymbol{\theta}}(\boldsymbol{s}))}$, where $V_{\boldsymbol{\theta}}(\boldsymbol{s}) = \max_{\boldsymbol{a}} Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$:

$$\begin{aligned} &\mathcal{L}^{\star}_{\boldsymbol{\theta}}(\boldsymbol{s}, \pi_{\text{prior}}, \lambda) \\ &= \frac{1}{\lambda} \log \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s}) \exp\left(\lambda Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})\right) \frac{\exp\left(\lambda V_{\boldsymbol{\theta}}(\boldsymbol{s})\right)}{\exp\left(\lambda V_{\boldsymbol{\theta}}(\boldsymbol{s})\right)} \\ &= V_{\boldsymbol{\theta}}(\boldsymbol{s}) + \frac{1}{\lambda} \log \sum_{\boldsymbol{a} \in \mathcal{A}} \pi_{\text{prior}}(\boldsymbol{a}|\boldsymbol{s}) \exp\left(\lambda \left(Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) - V_{\boldsymbol{\theta}}(\boldsymbol{s})\right)\right) \end{aligned}$$

The first term represents the ordinary maximum operator as in vanilla deep Q-learning. The second term is a log-partition sum with computationally stable elements due to the non-positive exponents $\lambda(Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a}) - V_{\boldsymbol{\theta}}(\boldsymbol{s})) \leq 0$. As a result, the log-partition sum is non-positive and subtracts a portion from $V_{\boldsymbol{\theta}}(\boldsymbol{s})$ that reflects how the instantaneous information-theoretic penalty translates into a penalty of cumulative reward values.

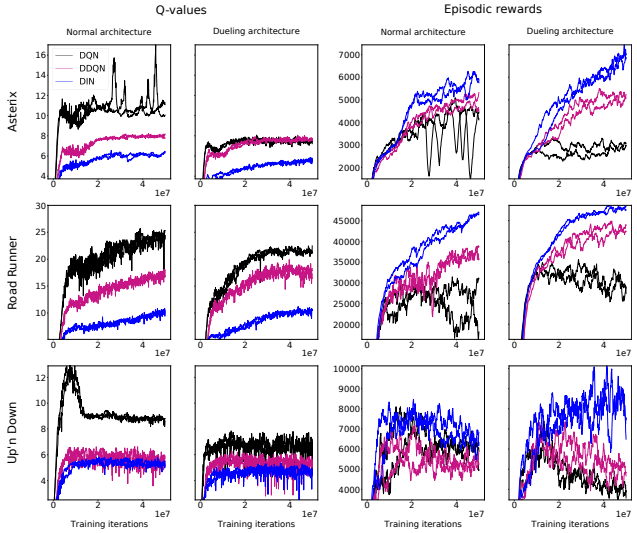

Figure 1. Q-values and episodic rewards for Asterix, Road Runner and Up'n Down for both normal and dueling architectures. Each plot shows three pairs of graphs, reporting the outcomes of two different random seeds, in black for DQN, purple for double DQN (DDQN) and blue for our information-theoretic approach (DIN). Clearly, our approach leads to lower Q-value estimates resulting in significantly better game play performance.

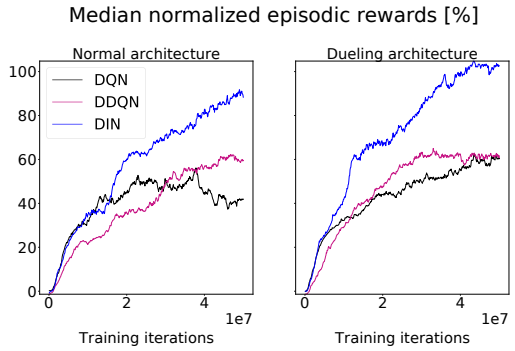

Figure 2. Median normalized episodic rewards across 20 Atari games for both normal and dueling architectures. Each plot compares DQN (black), against double DQN (DDQN, purple) and our approach (DIN, blue). Our approach leads to significantly higher median game score for both architectures.

5. Experiments & Results

We hypothesize that addressing the overestimation problem results in improved sample efficiency and overall performance. To this end, we use the Arcade Learning Environment (Bellemare et al., 2013) as a benchmark to evaluate our method. We compare against deep Q-networks (Mnih et al., 2015) that are susceptible to overestimations, and to double deep Q-networks (van Hasselt et al., 2016)—an alternative proposed to address the precise problem we target. Our results demonstrate that our proposed method (titled deep information networks DIN) leads to significantly lower Q-value estimates resulting in improved sample efficiency and game play performance. We also show that these findings remain valid for the recently proposed dueling architecture (Wang et al., 2016)[2].

5.1. Parameter Settings for Reproducibility

We conduct all experiments in Python with TensorFlow and OpenAI gym extending the GitHub project from (Kim, 2016). We use a deep convolutional neural network $Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$ as a function approximator for Q-values, designed and trained according to (Mnih et al., 2015). $Q_{\boldsymbol{\theta}}(\boldsymbol{s}, \boldsymbol{a})$ receives as input the current state of the environment $\boldsymbol{s}$ that is composed of the last four video frames. The number of neurons in the output layer is set to be the number of possible actions $\boldsymbol{a}$. Numerical values of each output neuron correspond to the expected cumulative reward when taking the relevant action in state $\boldsymbol{s}$.

We train the network for 5×10^7 iterations where one iteration corresponds to a single interaction with the environment. Environment interactions $(\boldsymbol{s}, \boldsymbol{a}, r, \boldsymbol{s}')$ are stored in a replay memory consisting of at most 10^6 elements. Every fourth iteration, a minibatch of size 32 is sampled from the replay memory and a gradient update is conducted with a discount factor $\gamma = 0.99$. We use RMSProp (Tieleman & Hinton, 2012) as the optimizer with learning rate 2.5×10^{-4}, gradient momentum 0.95, squared gradient momentum 0.95, and minimum squared gradient 0.01. Rewards r are clipped to $\{-1, 0, 1\}$. The target network $Q_{\boldsymbol{\theta}^-}(s, a)$ is updated every 10^4 iterations. The time constant τ for dynamically updating the hyperparameter λ is 10^5, and the prior policy π_{prior} is uniform. A uniform prior ensures a pessimistic baseline in case of small λ. This pessimistic baseline guarantees the existence of unbiased λ-configurations our scheduling scheme aims to detect.

When the agent interacts with the environment, every fourth frame is skipped and the current action is repeated on the skipped frames. During training, the agent follows an ϵ-greedy policy where ϵ is initialized to 1 and linearly annealed over 10^6 iterations until a final value of $\epsilon = 0.1$. Training and ϵ-annealing start at 5×10^4 iterations. RGB-images from the Arcade Learning Environment are preprocessed by taking the pixel-wise maximum with the previous image. After preprocessing, images are transformed to grey scale and down-sampled to 84×84 pixels. All our experiments are conducted in duplicate with two different initial random seeds. The random number of NOOP-actions at the beginning of each game episode is between 1 and 30.

We compare our approach against deep Q-networks and double deep Q-networks. In addition, we conduct further experiments by replacing network outputs with the dueling architecture (Wang et al., 2016). The dueling architecture leverages the advantage function $A(\boldsymbol{s}, \boldsymbol{a}) = Q(\boldsymbol{s}, \boldsymbol{a}) - \max_{\boldsymbol{a}} Q(\boldsymbol{s}, \boldsymbol{a})$ to approximate Q-values and generalizes learning across actions. This results in improved game play performance, as confirmed in our experiments.

5.2. Q-Values and Game Play Performance

During training, network parameters are stored every 10^5 iterations and used for offline evaluation. Evaluating a single network offline comprises 100 game play episodes lasting for at most 4.5×10^3 iterations. In evaluation mode, the agent follows an ϵ-greedy policy with $\epsilon = 0.05$ (Mnih et al., 2015). We investigate 20 Atari games.

Figure 1 reports results from the offline evaluation on three individual games (Asterix, Road Runner and Up'n Down), illustrating average maximum Q-values and average episodic

[2]Note that our approach can also be incorporated into the newly released Rainbow framework (Hessel et al., 2018) that achieves state-of-the-art results by combining several independent DQN improvements over the past few years (one of them being double DQNs over which our approach achieves superior performance). Although we focus on Q-value identification in this work, ideas similar to DIN do apply as well to actor-critic methods like A3C (Mnih et al., 2016; Schulman et al., 2017).

rewards as a function of training iterations. Note that episodic rewards are smoothed with an exponential window, similar to Equation (7) with $\tau = 10$, to preserve a clearer view. On all three games, our approach leads to significantly lower Q-value estimates when compared to DQN and double DQN for both, the normal and the dueling architecture (see left plots in Figure 1). At the same time, this leads to significant improvements in game play performance (see right plots of Figure 1).

Absolute episodic rewards (score) may vary substantially between different games. To ensure comparability across games, we normalize episodic rewards ($\text{score}_{\text{norm}}$) as

$$\text{score}_{\text{norm}} = \frac{\text{score} - \text{score}_{\text{random}}}{\text{score}_{\text{human}} - \text{score}_{\text{random}}} \cdot 100\%, \qquad (8)$$

where $\text{score}_{\text{random}}$ and $\text{score}_{\text{human}}$ refer to random and human baselines, see (Mnih et al., 2015; Wang et al., 2016).

Normalized episodic rewards enable a comparison across all 20 Atari games by taking the median normalized score over games (Hessel et al., 2018). The results of this analysis are depicted in Figure 2 as a function of training iterations (smoothed with an exponential window using $\tau = 10$). Our approach clearly outperforms DQN and double DQN for both normal and dueling architectures. The dueling architecture yields an additional performance increase when combined with DIN.

5.3. Sample Efficiency

To quantify sample efficiency, we identify the minimal number of training iterations required to attain maximum deep Q-network performance. To this end, we compute the average episodic reward as in Figure 1 but smoothed with an exponential window $\tau = 100$. We then identify for each approach the number of training iterations at which maximum deep Q-network performance is attained first.

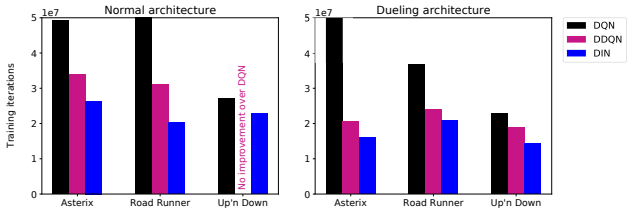

Figure 3. Sample efficiency for Asterix, Road Runner and Up'n Down under both normal and dueling architectures. There are three bars per game for DQN (black), double DQN (DDQN, purple) and our approach (DIN, blue). Clearly, DINs are more sample-efficient for both architectures on these three games.

The results for Asterix, Road Runner and Up'n Down are shown in Figure 3. It can be seen that our approach leads to significant improvements in sample efficiency when compared to DQN and double DQN. For instance, DINs require only about 2×10^7 training iterations in Road Runner compared to about 3×10^7 for double DQNs, and about 5×10^7 for standard DQNs using the normal architecture. These improvements are also valid for the dueling setting.

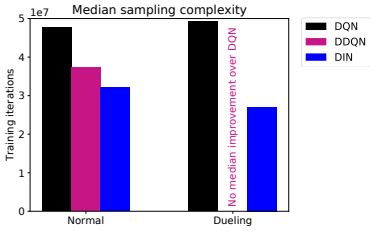

Figure 4. Median sample efficiency across 20 Atari games for both normal and dueling architectures. The plot compares DQN (black) against double DQN (DDQN, purple) and our approach (DIN, blue). DINs achieve significantly better median sample efficiency for both types of architecture.

In order to assess sample efficiency across all 20 Atari games, we compute the median sampling efficiency over games, see Figure 4. This analysis confirms the overall improved sample complexity attained in a wide range of tasks by our approach compared to DQN and double DQN.

Table 1. Normalized episodic rewards (normal architecture).

GAME	DQN	DDQN	DIN
ASSAULT	198.8%	214.9%	**233.5%**
ASTERIX	70.4%	73.4%	**85.0%**
BANK HEIST	76.2%	68.3%	**87.8%**
BEAMRIDER	123.6%	117.5%	**127.2%**
BERZERK	**35.8%**	33.7%	22.3%
DOUBLE DUNK	161.6%	**265.8%**	165.8%
FISHING DERBY	205.4%	**211.2%**	202.4%
FREEWAY	**103.3%**	75.2%	101.6%
KANGAROO	129.8%	94.4%	**137.1%**
KRULL	401.9%	494.8%	**534.6%**
KUNG FU MASTER	130.0%	-0.8%	**144.5%**
QBERT	22.1%	**31.9%**	21.4%
RIVERRAID	14.1%	20.4%	**26.1%**
ROAD RUNNER	503.6%	593.2%	**643.7%**
SEAQUEST	**4.0%**	1.2%	2.9%
SPACE INVADERS	53.9%	51.1%	**54.0%**
STAR GUNNER	560.5%	571.1%	**595.0%**
TIME PILOT	40.1%	**175.0%**	171.4%
UP'N DOWN	131.5%	**135.8%**	135.2%
VIDEO PINBALL	4385.8%	**5436.6%**	4654.1%
MEDIAN	126.7%	106.0%	**136.2%**

5.4. Policy Evaluation

We compare the performance of all approaches in terms of the best (non-smoothed) episodic reward (averaged over 100

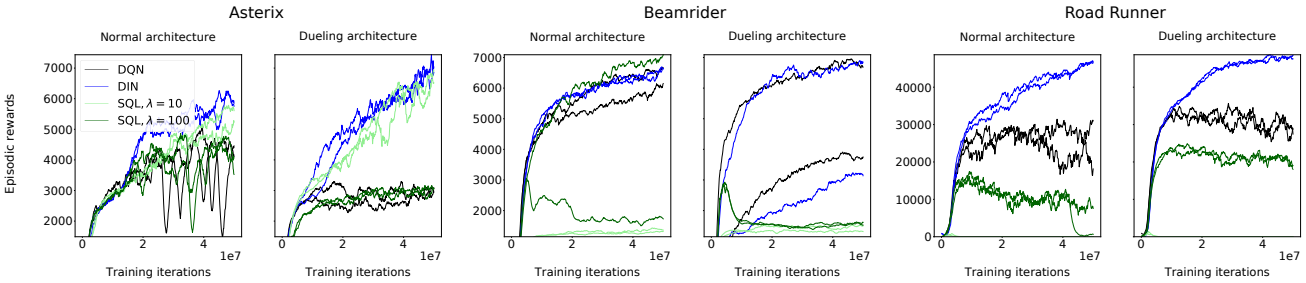

Figure 5. Episodic rewards for Asterix, Beamrider and Road Runner comparing our method to SQL. Clearly, our results show better performance in both the normal and dueling architecture without the necessity of identifying an optimal λ in advance.

Table 2. Normalized episodic rewards (dueling architecture).

GAME	DQN	DDQN	DIN
ASSAULT	260.4%	269.5%	**336.6%**
ASTERIX	45.6%	75.7%	**104.1%**
BANK HEIST	74.2%	78.5%	**79.1%**
BEAMRIDER	**130.6%**	128.4%	129.3%
BERZERK	32.5%	**33.4%**	24.6%
DOUBLE DUNK	223.9%	**241.6%**	222.6%
FISHING DERBY	**177.2%**	163.4%	29.3%
FREEWAY	103.8%	102.9%	**104.6%**
KANGAROO	347.6%	186.8%	**437.4%**
KRULL	480.8%	433.6%	**574.3%**
KUNG FU MASTER	114.8%	118.4%	**129.6%**
QBERT	70.2%	**84.2%**	82.8%
RIVERRAID	**28.4%**	22.9%	22.8%
ROAD RUNNER	553.6%	624.4%	**659.0%**
SEAQUEST	**14.4%**	0.5%	5.1%
SPACE INVADERS	63.1%	26.7%	**140.6%**
STAR GUNNER	139.5%	145.1%	**169.4%**
TIME PILOT	109.6%	56.7%	**265.1%**
UP'N DOWN	105.5%	125.6%	**150.9%**
VIDEO PINBALL	4461.6%	4754.5%	**4982.8%**
MEDIAN	112.2%	122.0%	**135.1%**

episodes) obtained in the course of the entire evaluation procedure. To ensure comparability between games, we again make use of normalized scores according to Equation (8).

Our results are summarized for the normal and dueling architecture in Tables 1 and 2 respectively. In both cases, our approach achieves superior median normalized game performance compared to DQN and double DQN. In the normal setting, DIN achieves best performance across all three approaches in 11 out of 20 games, whereas in the dueling setting, DIN achieves best performance in 13 out of 20 games. Note that we can confirm that the dueling architecture, when combined with DIN, leads to a performance increase in 15 out of 20 games, which is however not reflected in the median performance.

5.5. Comparison to Soft-Q Learning (SQL)

As mentioned earlier, the closest work to our approach is that of (Schulman et al., 2017), where the authors consider information theory to bridge the gap between Q-learning and policy gradients RL. Our approach goes further by considering dynamic adaptation for λ in the course of training, and introduces robust computation based on value advantages. We compare our method to SQL (where λ is fixed) on the games Asterix, Beamrider and Up'n Down. Results depicted in Figure 5 demonstrate that our method can outperform SQL on these three games by significant margins without the requirement of pre-specifying λ. For instance, DINs achieve the best performance of SQL in about 5,000,000 iterations on the Road Runner game.

6. Conclusions & Future Work

In this paper, we proposed a novel method for reducing sample complexity in deep reinforcement learning. Our technique introduces an intrinsic penalty signal by adapting principles from information theory to high-dimensional state spaces. We showed that DQNs are a special case of our proposed approach for a specific choice of the Lagrange multiplier steering the intrinsic penalty. Finally, in a set of experiments on 20 Atari games, we demonstrated that our technique indeed outperforms competing approaches in terms of performance and sample efficiency. These results remain valid for the dueling architecture from (Wang et al., 2016) yielding a further performance boost.

The most promising direction of future work is to study adaptive prior policies instead of fixed ones, in line with (Bagnell & Schneider, 2003; Peters & Schaal, 2008; Peters et al., 2010; Schulman et al., 2015). This could be used to extend our framework to multi-task learning scenarios where task-specific policies satisfy a KL-constraint to prevent deviation from a common prior. The common prior can encode a behavioral policy that generalizes across tasks, thus enabling knowledge transfer between problems with a shared latent structure.

References

Azar, M G, Munos, R, Ghavamzadeh, M, and Kappen, H J. Speedy Q-learning. In *Advances in Neural Information Processing Systems*, 2011.

Azar, M G, Gomez, V, and Kappen, H J. Dynamic policy programming. *Journal of Machine Learning Research*, 13:3207–3245, 2012.

Bagnell, J A and Schneider, J. Covariant policy search. *Proceedings of the International Joint Conference on Artificial Intelligence*, 2003.

Bellemare, M G, Naddaf, Y, Veness, J, and Bowling, M. The Arcade Learning Environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Bellemare, M G, Ostrovski, G, Guez, A, Thomas, P S, and Munos, R. Increasing the action gap: New operators for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

Busoniu, L, Babuska, R, De Schutter, B, and Ernst, D. *Reinforcement Learning and Dynamic Programming using Function Approximators*. CRC Press, 2010.

Csiszar, I and Tusnady, G. Information geometry and alternating minimization procedures. *Statistics and Decisions*, (Supplement 1):205–237, 1984.

Fox, R, Pakman, A, and Tishby, N. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2016.

Genewein, T, Leibfried, F, Grau-Moya, J, and Braun, D A. Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle. *Frontiers in Robotics and AI*, 2(27), 2015.

Grau-Moya, J, Leibfried, F, Genewein, T, and Braun, D A. Planning with information-processing constraints and model uncertainty in Markov decision processes. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 2016.

Haarnoja, T, Tang, H, Abbeel, P, and Levine, S. Reinforcement learning with deep energy-based policies. *Proceedings of the International Conference on Machine Learning*, 2017a.

Haarnoja, T, Zhou, A, Abbeel, P, and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Advances in Neural Information Processing Systems*, 2017b.

Hessel, M, Modayil, J, van Hasselt, H, Schaul, T, Ostrovski, G, Dabney, W, Horgan, D, Piot, B, Azar, M, and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Kappen, H J, Gomez, V, and Opper, M. Optimal control as a graphical model inference problem. *Machine Learning*, 87(2):159–182, 2012.

Kim, T. Deep reinforcement learning in TensorFlow, 2016. URL https://github.com/carpedm20/deep-rl-tensorflow.

Lee, D and Powell, W B. An intelligent battery controller using bias-corrected Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2012.

Leibfried, F and Braun, D A. A reward-maximizing spiking neuron as a bounded rational decision maker. *Neural Computation*, 27(8):1686–1720, 2015.

Leibfried, F and Braun, D A. Bounded rational decision-making in feedforward neural networks. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2016.

Lin, L-J. *Reinforcement learning for robots using neural networks*. PhD thesis, Carnegie Mellon University, 1993.

Mnih, V, Kavukcuoglu, K, Silver, D, Rusu, A A, Veness, J, Bellemare, M G, Graves, A, Riedmiller, M, Fidjeland, A K, Ostrovski, G, Petersen, S, Beattie, C, Sadik, A, Antonoglou, I, King, H, Kumaran, D, Wierstra, D, Legg, S, and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Mnih, V, Puigdomenech Badia, A, Mirza, M, Graves, A, Lillicrap, T P, Harley, T, Silver, D, and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2016.

Nachum, O, Norouzi, M, Xu, K, and Schuurmans, D. Bridging the gap between value and policy based reinforcement learning. *Advances in Neural Information Processing Systems*, 2017.

O'Donoghue, B, Munos, R, Kavukcuoglu, K, and Mnih, V. Combining policy gradient and Q-learning. *Proceedings of the International Conference on Learning Representations*, 2017.

Ortega, P A and Braun, D A. Thermodynamics as a theory of decision-making with information-processing costs. *Proceedings of the Royal Society A*, 469(2153), 2013.

Peng, Z, Genewein, T, Leibfried, F, and Braun, D A. An information-theoretic on-line update principle for perception-action coupling. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

Peters, J and Schaal, S. Reinforcement learning of motor skills with policy gradients. *Neural Networks*, 21:682–697, 2008.

Peters, J, Mulling, K, and Altun, Y. Relative entropy policy search. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2010.

Rawlik, K, Toussaint, M, and Vijayakumar, S. On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings Robotics: Science and Systems*, 2012.

Rubin, J, Shamir, O, and Tishby, N. Trading value and information in MDPs. In *Decision Making with Imperfect Decision Makers*, chapter 3. Springer, 2012.

Schulman, J, Levine, S, Moritz, P, Jordan, M, and Abbeel, P. Trust region policy optimization. In *Proceedings of the International Conference on Machine Learning*, 2015.

Schulman, J, Abbeel, P, and Chen, X. Equivalence between policy gradients and soft Q-learning. *arXiv*, 1704.06440, 2017.

Shannon, C E. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423,623–656, 1948.

Sims, C A. Rational inattention and monetary economics. In *Handbook of Monetary Economics*, volume 3, chapter 4. Elsevier, 2010.

Sutton, R S and Barto, A G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Thrun, S and Schwartz, A. Issues in using function approximation for reinforcement learning. In *Proceedings of the Connectionist Models Summer School*, 1993.

Tieleman, T and Hinton, G E. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. In *COURSERA: Neural Networks for Machine Learning*, 2012.

Tishby, N and Polani, D. Information theory of decisions and actions. In *Perception-Action Cycle*, chapter 19. Springer, 2011.

Tishby, N, Pereira, F C, and Bialek, W. The information bottleneck method. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*, 1999.

Todorov, E. Efficient computation of optimal actions. *Proceedings of the National Academy of Sciences of the United States of America*, 106(28):11478–11483, 2009.

van Hasselt, H. Double Q-learning. In *Advances in Neural Information Processing Systems*, 2010.

van Hasselt, H, Guez, A, and Silver, D. Deep reinforcement learning with double Q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2016.

Wang, Z, Schaul, T, Hessel, M, van Hasselt, H, Lanctot, M, and de Freitas, N. Dueling network architectures for deep reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2016.

Watkins, C J C H. *Learning from delayed rewards*. PhD thesis, University of Cambridge, 1989.

Williams, R J and Peng, J. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.