Boosting Soft Q-Learning by Bounding

Jacob Adamczyk
jacob.adamczyk001@umb.edu
Department of Physics
University of Massachusetts Boston

Volodymyr Makarenko
volodymyr.makarenko@sjsu.edu
Department of Computer Engineering
San José State University

Stas Tiomkin
stas.tiomkin@sjsu.edu
Department of Computer Engineering
San José State University

Rahul V. Kulkarni
rahul.kulkarni@umb.edu
Department of Physics
University of Massachusetts Boston

Abstract

An agent's ability to leverage past experience is critical for efficiently solving new tasks. Prior work has focused on using value function estimates to obtain zero-shot approximations for solutions to a new task. In soft Q-learning, we show how any value function estimate can also be used to derive double-sided bounds on the optimal value function. The derived bounds lead to new approaches for boosting training performance which we validate experimentally. Notably, we find that the proposed framework suggests an alternative method for updating the Q-function, leading to boosted performance.

1 Introduction

In recent years, reinforcement learning (RL) has seen impressive success at the price of ever-increasing sample budgets. The current paradigm of RL consists of training agents from scratch with new hyperparameters or in new domains, without significant reuse of previously collected information. The large datasets generated from such runs have been approached with techniques such as offline RL; however, the approximate solutions obtained from previous runs are typically not reused. To address this issue, approaches that directly leverage this learned prior knowledge to efficiently calculate policies for new tasks are needed. While prior solutions may not be *optimal* for arbitrary new tasks, they have been shown to serve as useful approximations that reduce training time in a variety of settings: (Rusu et al., 2016; Tasse et al., 2021; Agarwal et al., 2022; Adamczyk et al., 2023b). In this work, we present a new way in which information from previous solutions can be further leveraged to address new tasks[1].

Previous work has focused on addressing this challenge with approaches such as transfer learning, curriculum learning, and compositionality. In this work, we will focus on value-based RL algorithms where the agent learns the optimal action-value, or Q-function. In many instances, the agent has an estimate of the value function even before training begins. For example, in the case of curriculum learning, the agent has the Q-values for previously learned (progressively more challenging) tasks. In the case of compositional (Haarnoja et al., 2018a) or hierarchical RL (Hafner et al., 2022), the agent can combine knowledge by applying a function on the subtasks' Q-values. When using an exploratory skill-acquisition approach (Eysenbach et al., 2019) or constructing a task basis (Alver & Precup, 2021), the agent obtains solutions for a diverse set of skills to use on downstream tasks. Even in cases where an initial estimate is not explicitly provided, the agent indeed has access to an estimate through the Q-values obtained in the ongoing learning phase (bootstrapping).

[1] Our code is publicly available at https://github.com/JacobHA/RLC-SoftQBounding.

An underlying question in these scenarios is the following: How can the agent use these known value function estimate(s) for solving a new target task? Does the estimate only serve as a zero-shot approximation or is there additional useful information that can be extracted from such estimates? **In this work, we address this question by deriving bounds on the *optimal* Q-values from an *arbitrary* prior estimate.** We emphasize that the surprising nature of these bounds is that information concerning the *optimal* value function can be derived from arbitrarily *suboptimal* estimates.

To derive such bounds, we leverage exact results on the Q-function from recent work by Cao et al. (2021) and Adamczyk et al. (2023a). In the latter, the authors show (in their Theorem 1) that there exists a method of "closing the gap" between any estimate (therein denoted $Q^*(s,a)$) and any target (denoted $\widetilde{Q}^*(s,a)$) task in entropy-regularized RL. Here, we show that since this gap between the target and estimated value functions, $\widetilde{Q}^*(s,a) - Q^*(s,a) \;=\; K^*(s,a)$, is itself an optimal value function, it can be bounded. As a consequence, instead of providing only a zero-shot approximation ("warmstart" or "jumpstart") for training the target task, we show that the estimates available to the agent also provide a double-sided bound on the desired optimal Q-values. From Theorem 1 of Cao et al. (2021), it can be shown that an optimal solution is not required for deriving such bounds, and in fact *any* function over the state-action space can be used to derive double-sided bounds, including for instance the bootstrapped estimate of $Q(s,a)$. Since it is the most general, we focus on this case in the present work.

A schematic illustration of our approach is provided in Figure 1. Starting with samples of a value function (red points), we derive double-sided bounds (dashed blue lines) on the optimal value function (solid black line). We find that applying these bounds during training significantly boosts the agent's training performance in the tabular setting. We provide further theoretical analysis in continuous state-action spaces, relevant for the function approximator (FA) setting in deep reinforcement learning, for which we present initial experiments in Section 5.

Main contributions
The main contributions of our work are as follows:

1. A framework for bounding optimal value functions given any estimate of the value function.
2. A novel soft Q-learning algorithm and demonstration of its advantages.
3. Extension of theoretical results to continuous state-action spaces.

2 Preliminaries

For the theoretical analysis, we begin with finite, discrete state and action spaces, and we subsequently extend our analysis to continuous spaces. In this setting, the RL problem is modeled as a Markov Decision Process (MDP) represented by the tuple $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma \rangle$ where $\mathcal{S}$ is the state space; $\mathcal{A}$ is the action space; $p : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function (dynamics); $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a (bounded) reward function; and $\gamma \in [0, 1)$ is the discount factor. We focus on the generalization of entropy-regularized RL (Ziebart, 2010), which augments the un-regularized RL objective by including an entropic regularization term which penalizes control over a pre-specified reference policy:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \left(r_t - \frac{1}{\beta} \log\left(\frac{\pi(a_t|s_t)}{\pi_0(a_t|s_t)} \right) \right)\right]$$

where $\pi_0(a|s)$ is a fixed prior policy. This additional control cost discourages the agent from choosing policies that deviate too much from the prior policy. Importantly, entropy-regularized MDPs lead to stochastic optimal policies that are provably robust to perturbations of rewards and dynamics (Eysenbach & Levine, 2022); making for a more suitable approach to real-world problems.

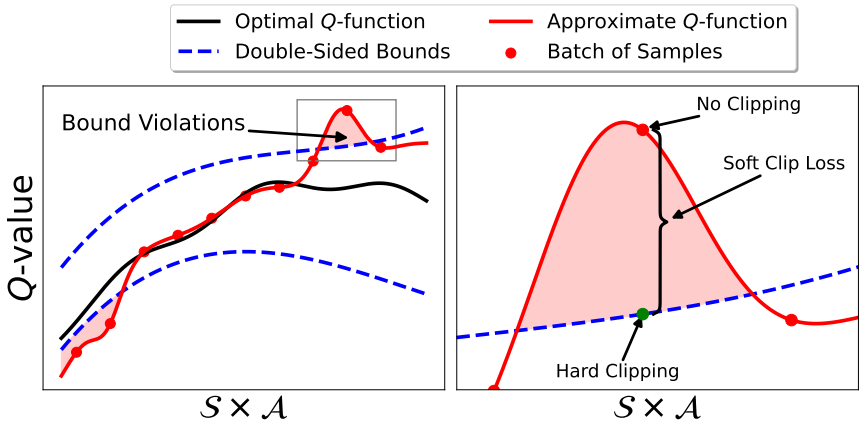

Figure 1: Schematic illustration of the main contribution of this work. Given any approximation (red curve) to the optimal value function of interest (black curve), we derive double-sided bounds (blue curves) that lead to clipping approaches during training. Based solely on the current approximation for $Q(s, a)$ (red curve), we derive double-sided bounds on the *unknown* optimal value function $Q^*(s, a)$ (black curve). In the right panel, we show the different clipping methods, which are described further in the "Experimental Validation" section. In "Hard Clipping", the target is replaced with the exceeded bound; in "Soft Clipping", an additional loss term is appended to the Bellman loss, proportional to the magnitude of the bound violation.

The solution to the RL problem is defined by its optimal action-value function ($Q^*(s, a)$) from which one can derive the aforementioned optimal policy $\pi^*(a|s)$ through a Boltzmann distribution with temperature β^{-1}. The optimal value function can be obtained by iterating the following recursive Bellman equation (Ziebart, 2010; Haarnoja et al., 2018b):

$$Q^*(s, a) = r(s, a) + \frac{\gamma}{\beta} \mathop{\mathbb{E}}_{s' \sim p} \log \mathop{\mathbb{E}}_{a' \sim \pi_0} e^{\beta Q^*(s', a')}. \tag{1}$$

The regularization parameter β is used to control the degree of stochasticity in the optimal policy. In the entropy-regularized setting, Q^* is referred to as the optimal "soft" action-value function. For brevity, we refer to Q^* simply as the value function when context is clear.

3 Prior Work

The use of value function bounds has been investigated in various domains of RL: offline and online settings, compositionality, and imitation learning. In this section, we briefly outline some of the most relevant work from this domain. We contrast the existing literature with regard to the following features: i) the MDP's structural assumptions, ii) the requirement for additional samples to derive bounds, iii) use of double or single-sided bounds.

In (Nemecek & Parr, 2021), the authors have derived double-sided bounds on the state value function $V(s)$ when the task's reward function can be written as the *positive conical combination* of subtask rewards. This method requires additional samples for first learning the *successor features* (SFs) before then deriving the double-sided bounds for a downstream task. The aforementioned work was subsequently extended by Kim et al. (2022), where, in the same SF setting, they present double-sided bounds on Q-values for linear combinations of subtask reward functions. They introduced a notion of "soft clipping" but it was not demonstrated in practice. We later adapt this idea of soft clipping to our setting, with details in Section 5. Both Nemecek & Parr (2021) and Kim et al. (2022) consider the un-regularized RL problem formulation ($\beta \to \infty$).

The two previous methods focus on the standard (un-regularized) reinforcement learning setting. However, the double-sided bounds presented by Haarnoja et al. (2018a)'s Lemma 1 are derived for

the MaxEnt setting, for the case of convex reward combinations. It is worth noting that the lower bound in this case must be learned (their C function). Extending these results to other more general classes of functional composition, Adamczyk et al. (2023b) provides double-sided bounds for both entropy-regularized and un-regularized RL. However, one side of the bounds in all cases must be learned as well.

In a different context focused on the stability of value-based RL, Lee et al. (2021) proposes to (approximately) bound Bellman updates through a weighted ensemble, which improves the stability of training and sample efficiency in entropy-regularized RL. However, the method by Lee et al. (2021) cannot leverage known solutions for new tasks, instead using a parallel ensemble of learners for variance estimation, exploiting the UCB framework (Auer et al., 2002) for exploration bonuses.

Other examples of deep RL utilizing bounds include (He et al., 2017), which utilize bounds based on n-step returns, resulting in faster reward accumulation in the Atari suite. However, their bounds were not tested in stochastic environments but were shown to hold in expectation. Further, their upper bound depends on Q^*, the unknown optimal value function, making it intractable without first solving the task in question. Later, the work of Hoppe & Toussaint (2020) modeled Q-functions through graphical models, using this structure to derive various bounds used in a constrained-DDPG algorithm. In this algorithm, a "hard clipping" mechanism is used, wherein the updates to the Q-values were clipped based on their bounds. We will consider a similar hard clipping approach, discussed in Section 5, but we derive bounds without imposing a graphical model of the dynamics. In principle, our bounds can interface easily with such prior work, by straightforwardly combining methods to obtain the tightest bounds possible.

In contrast to the aforementioned work, we propose a novel method for calculating double-sided bounds, not limited to a particular type of composition of prior solution(s) and valid for an *arbitrary* input function. Our method for deriving double-sided bounds is zero-shot — it does not require additional samples beyond those collected by the learning agent. Furthermore, our results are applicable to stochastic environments and both discrete or continuous domains.

4 Results

Our main result provides double-sided bounds on the optimal Q-function. We emphasize that *any* (bounded) function $Q : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ can be used to generate such bounds. We suggestively use the notation "$Q(s, a)$" for this otherwise arbitrary function to emphasize that it may be derived from a previous task's solution, an estimate, or other ansatz (e.g. composition or hierarchical function) of Q-values.

Theorem 1. *Consider an entropy-regularized MDP $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma, \beta, \pi_0 \rangle$ with optimal value function $Q^*(s, a)$. Let any bounded function $Q(s, a)$ be given. Denote the corresponding state-value function as $V(s) \doteq 1/\beta \log \mathbb{E}_{a\sim\pi_0} \exp \beta Q(s, a)$. Then, $Q^*(s, a)$ is bounded by:*

$$r(s,a) + \gamma \left(\mathop{\mathbb{E}}_{s'\sim p} V(s') + \frac{\inf \Delta}{1-\gamma} \right) \leq Q^*(s,a) \leq r(s,a) + \gamma \left(\mathop{\mathbb{E}}_{s'\sim p} V(s') + \frac{\sup \Delta}{1-\gamma} \right) \tag{2}$$

where

$$\Delta(s,a) \doteq r(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim p} V(s') - Q(s,a).$$

In Equation 2, the inf and sup are taken over the (potentially continuous) state-action space $\mathcal{S} \times \mathcal{A}$. The proof of Theorem 1 can be found in Appendix B. Intuitively, this result can be understood as a double-sided bound on the *optimal* Q-function, calculated through a single iterate of the Bellman operator (B) on *any* input function (denoted Q):

$$\left| Q^*(s,a) - BQ(s,a) \right| \leq \mathcal{O}\left(H\sqrt{\mathcal{L}} \right), \tag{3}$$

where $H = (1-\gamma)^{-1}$ denotes the effective time horizon and $\mathcal{L}$ denotes the Bellman loss incurred by the input function Q. During training, the Bellman loss (ideally) reduces to zero: $\mathcal{L} = ||\Delta||^2 \to 0$, implying that $\inf \Delta \to 0$ and $\sup \Delta \to 0$, hence the bounds in Theorem 1 are tight upon convergence of the soft action-value function.

As an alternative, in practice, one can replace the inf and sup in the previous results by a min and max, respectively, over some finite dataset (e.g. the current batch of replay data). Although not exact, this substitution becomes increasingly accurate for large datasets (batch sizes), as formalized by our Theorem 2 (Informal). We employ this substitution in the function approximator experiments shown in Section 5.

After calculating (or estimating) the lower and upper bounds in Theorem 1, we propose to clip the Q-function with these bounds at each training step. We conclude this section by showing that the Bellman operator with clipping converges to the optimal Q-function:

Proposition 1. *Let $B(\cdot)$ denote the Bellman operator, and let the functions $L(s,a)$, $U(s,a)$ be lower and upper bounds on the optimal action-value function respectively: $L(s,a) \leq Q^*(s,a) \leq U(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The clipped Bellman operator, $B_C Q(s,a) := \max\left(\min\left(BQ(s,a), U(s,a)\right), L(s,a)\right)$ converges to the optimal action-value function $Q^*(s,a) = B^\infty Q(s,a) = B_C^\infty Q(s,a)$ for any bounded initial function $Q(s,a)$.*

This result shows that updates with and without clipping are guaranteed to converge to the same fixed point, $Q^*(s,a)$ (proof in Appendix B). We experimentally demonstrate this statement in Figure 2.

4.1 Extension to Continuous Spaces

The bounds presented in the previous section, though exact, are often intractable due to the required global extremization over continuous state-action spaces. We therefore loosen the previous bounds by relaxing the required extremization with a simpler optimization over a given batch of replay data. To this end, we apply the results on Pure Random Search from (Malherbe & Vayatis, 2017), bounding the error of estimated extrema of Lipschitz-continuous functions in bounded continuous spaces. We next give a brief discussion on the required steps in the proof, and a full discussion and derivation of Theorem 2 are given in Appendix C. We will assume that the extrema of the MDP's reward function $r(s,a)$ can be estimated with high accuracy (since in principle, the entire replay buffer can be used). Instead, we will focus on the larger errors in Δ, which change over the course of training (since we use the most recently learned Q-values to generate bounds) and hence has a much smaller dataset available, e.g. the sampled replay batch, for estimation.

Two issues must be addressed before we can apply the concentration results from (Malherbe & Vayatis, 2017) to Theorem 1: (1) Their concentration analysis requires a Lipschitz constant for the function in question (Δ) which is not readily available; and (2) the exact value of Δ cannot be given in general, since the dependence on $V(s)$ requires computing an expectation value over states and actions. We surmount these two issues by first providing a calculation for the Lipschitz constant of Δ, based on the Lipschitz constant for a general soft Q-function. The derivation and discussion of this Lipschitz constant is given in Appendix C.2 due to space constraints. Secondly, in the case of stochastic transitions with continuous state-action spaces, we cannot calculate the state-value function term directly. Instead, we apply a concentration inequality to this expectation term, allowing us to bound its error with high probability. A similar error propagation must be considered from the sampling the prior policy in the continuous action setting (to approximate the action expectation in Equation 1). Hence, with sufficient smoothness and sampling hypotheses, we extend Theorem 1 to the sample-based case relevant for the FA setting:

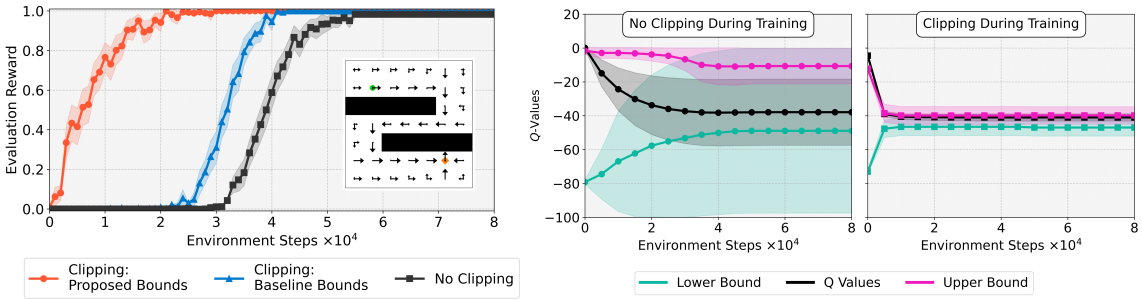

Figure 2: Here we show specific results on a representative environment, and further examples are given in the Appendix. At each step, the agent receives a small penalty if it has not reached the goal (orange diamond). The discount factor $\gamma = 0.98$ and inverse temperature parameter $\beta = 5$ are fixed throughout these experiments. From left to right: (1) The optimal policy is shown at the inset. The greedy policy is evaluated during training for the various methods presented. "Baseline Bounds" refers to clipping during training with $\left[\frac{r_{\min}}{1-\gamma}, \frac{r_{\max}}{1-\gamma}\right]$. (2,3) The mean and range of Q-values and the proposed bounds (Equation 2). Clipping during training constrains the Q-values to a tight range much faster than without clipping. Each method is averaged over 30 random initializations.

Theorem 2 (Informal). *Consider an MDP with a bounded continuous state and action space, $\mathcal{S} \times \mathcal{A} \subset \mathbb{R}^d$, with stochastic dynamics. Suppose an L_Q-Lipschitz function $Q(s,a)$ is given to generate double-sided bounds on the optimal value function, denoted $Q^*(s,a)$. Let $\varepsilon > 0, \delta > 0$ be given and define the horizon $H = (1-\gamma)^{-1}$, and sample budgets: $|\mathcal{B}| \geq \mathcal{O}\left(\varepsilon^{-d} \log \delta^{-1}\right)$, $n_{\mathcal{S}} \geq \mathcal{O}\left(H^2 \varepsilon^{-2} \log \delta^{-1}\right)$, $n_{\mathcal{A}} \geq \mathcal{O}\left(e^{2\beta(H-\varepsilon)} \log \delta^{-1}\right)$. Suppose $n_{\mathcal{S}}$ samples are used to estimate the expectation over next-states and $n_{\mathcal{A}}$ samples are used to estimate the expectation over next-actions in the soft state-value function. Denoting $\hat{V}, \hat{\Delta}$ as the quantities estimated from samples, the following bounds*

$$Q^*(s,a) \leq r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} \hat{V}(s_i') + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \varepsilon}{1-\gamma}\right) \tag{4}$$

$$Q^*(s,a) \geq r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} \hat{V}(s_i') + \frac{\min_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) - \varepsilon}{1-\gamma}\right) \tag{5}$$

hold with probability at least $1-\delta$.

This result shows that our bounds remain valid in the continuous state-action setting with stochastic dynamics, given sufficiently many samples (large batch sizes). We provide results for other scenarios (discrete or continuous states, deterministic or stochastic transition dynamics), as well as the formal result and relevant definitions in Appendix C.

5 Experimental Validation

In our experiments, we study the utility of clipping based on our theoretical results. For simplicity, we first highlight the results in discrete environments with tabular soft Q-learning. Without any external estimates for the Q-function, we use the estimate given by the previous step's Q-values. Note that this method of obtaining an estimate from the previous training step is the most general case, applicable to any value-based algorithm. If available, further information based on other solutions or task structure can additionally be used. Secondly, we study the extension of our theory

to the case of continuous states, using the same method for deriving bounds, where we compare clipping methods on the classic control benchmark.

We observe that the use of bounds for clipping the Q-function during training leads to a different training dynamics than the standard TD update. Intuitively, clipping restricts the Q-function *away* from invalid regions while the Bellman updates pull the Q-function *toward* the correct values. The state-action dependency of our bounds also seems to be a key feature, based on comparison to using the looser but constant "Baseline" bounds: $Q^*(s,a) \in \left[\frac{R_{\min}}{1-\gamma}, \frac{R_{\max}}{1-\gamma}\right]$ (cf. Figures 2 and 3).

5.1 Tabular Experiments

In the tabular case, since we have access to the Q-table, we can simply clip the updated Q-table according to the derived bounds. When the model (reward and transition table) is given, we maintain the lower and upper bounds throughout training, tightening them whenever a better bound becomes available. In Figure 2 we show the results of training in a simple maze environment. In the main plots of Figure 2, we depict a comparison of the evaluation rewards and the mean Q-value over all (s,a) pairs. In experiments across different sized environments, and with various levels of stochasticity, we universally find the increase in convergence speed shown.

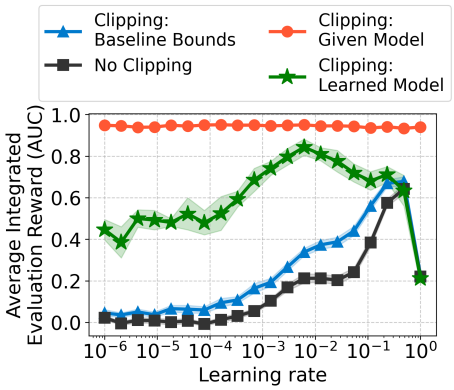

Figure 3: Speed of learning (measured as area under evaluation reward curve) with Q-value clipping during TD updates. Each point is the result of averaging over 30 randomly generated 7×7 mazes with stochastic transitions. Further details of the experiment are given in Appendix A.1.

To verify the robustness of clipping benefits, we sweep over the learning rate and maze topology, and measure the speed of convergence: plotted in Figure 3. The metric used is a normalized area under the evaluation reward curve. Since randomly generated mazes are used, we normalize against the performance of a uniform policy and the greedy optimal policy (given by the exact solution). Thus an algorithm which quickly obtains (and maintains) the optimal reward will have a larger success metric (more details in Appendix A.1). In this experiment, we use stochastic transition dynamics. We begin by giving the agent the model (in this case, simply a table of rewards and stochastic transition probabilities), which is required for an exact calculation of the bounds in Theorem 1. This setting of a "Given Model" is shown with red circles in Figure 3, which outperforms the other methods, for all learning rates. Then, we consider the case of a learned model, shown with green stars in Figure 3. This case represents a stepping stone from exact tabular updates (red line) to the function approximator case since there is noise in the calculation of $\Delta(s,a)$ based on sampling errors introduced by the learned model.

We have found that there are initializations for which our bounds are not violated and thus the Q-function is not (initially) clipped by our bounds. For instance, we find that for Q-values initialized far from zero, the bounds are loose and not violated, hence small learning rates will yield nearly static training dynamics which do not converge. This observation, coupled with the finding that updating via clipping (when the bounds *are* violated) consistently performs better (red line with circle markers in Fig. 3) leads us to the following: To take advantage of the boosting given by clipping in the low learning rate regime, we propose to use the standard temporal difference (TD) update only if the Q-function has not changed between two iterations (i.e. no clipping occurs), and otherwise clip the Q-values appropriately *without any TD update*. This should be distinguished from a simpler "always clip" approach, shown in Algorithm 1. This change ensures convergence of a tabular SQL algorithm while maximizing the utility of clipping. Pseudocode for the algorithm is given in Algorithm 2 in Appendix A.1. The performance difference between Algorithms 1 and 2 is

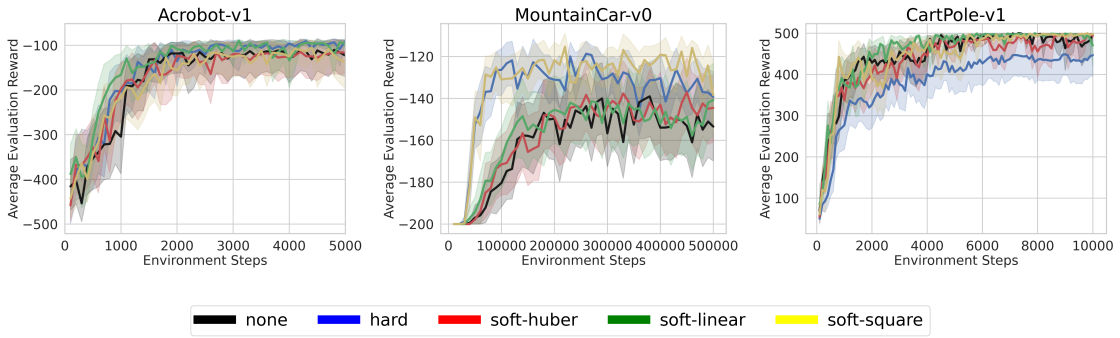

Figure 4: We test the proposed clipping methods (labeled None, Hard, and Soft; described below) across the classic control suite. We fine-tuned each environment's hyperparameters (details in Appendix A.1). The average evaluation reward plotted is the reward achieved by following the stochastic optimal policy, averaged over 5 episodes. Each method in a given environment is averaged over 30 random initializations, with the 95% bootstrapped confidence interval shaded. To ensure the performance stems from our bounds alone, we have not included the simpler $R_{\min,\max}/(1-\gamma)$ bounds which are likely to improve the performance further.

shown in Figure 6 in Appendix A. Two versions (one with the model given and one with a learned model) of our proposed algorithm are shown in Figure 3.

5.2 Function Approximator Experiments

To test our bounds in the deep RL setting, we turn to environments with continuous state spaces, adopting a DQN-style implementation of discrete-action soft Q-learning with an entropy-regularized TD update, using an online and target network. Although we have derived bounds for this case, we cannot simply clip the entire Q-function as we did in the tabular setting. The proposed algorithm (with clipping only if the Q-function remains fixed) cannot be directly translated to the FA setting. Thus, we will propose two different methods of clipping suitable for function approximators.

Since the soft Q-learning (SQL) algorithm employs a target network we use both the target and online networks to derive bounds on the optimal Q-values (cf. Appendix A for the implementation details and hyperparameters used for training). Since the bounds must hold when either the target or online net is used as an estimate, we can always take the tighter bound (s,a)-wise between the two. In general, given many sources of Q-function estimates (such as in ensemble methods), one can use them collectively to obtain the tightest bound possible.

The derived bounds can be implemented using different approaches for clipping of the value function during training. We highlight the different methods used below, inspired by the methods used by He et al. (2017); Kim et al. (2022); Adamczyk et al. (2023b):
(0) **No Clipping:** The standard training scheme for SQL is implemented.
(1) **Hard Clipping:** At each backup calculation we enforce the following bounds on the new Q-value:

$$Q(s,a) \leftarrow \hat{Q}_{\text{clip}}(s,a) \doteq \min\left\{\max\left\{r(s,a) + \frac{\gamma}{\beta}\log \mathop{\mathbb{E}}_{a'\sim\pi_0} \exp \beta Q(s',a'),\ \mathrm{L}(s,a)\right\}, \mathrm{U}(s,a)\right\} \tag{6}$$

and L and U denote the lower and upper bounds derived in Theorem 1. In the tabular setting, L and U can be calculated exactly. However, for function approximator experiments with sampling, we replace the inf and sup with a min and max over the current batch as justified in Section 4.
(2) **Soft Clipping:** An additional term, the "clipping loss", is added to the function approximator's loss function. The clipping loss is defined as

$$\mathcal{L}_{\text{clip}} = \frac{1}{|\mathcal{B}|}\sum \left|Q(s,a) - \hat{Q}_{\text{clip}}(s,a)\right|, \tag{7}$$

with a summation over the current batch, where $|\mathcal{B}|$ is the batch size. This gives a total loss of $\mathcal{L} = \mathcal{L}_{\text{Bellman}} + \eta \mathcal{L}_{\text{clip}}$. The hyperparameter η weights the relative importance of the bound violations against the Bellman error, whose value we fix to unity for simplicity. On an environment-wise basis, we fine-tune the hyperparameters on the baseline method (values in Appendix A.2) and use those same values for all clip methods. Figure 4 indicates that clipping can lead to improvements in the speed of training; however, additional modifications are needed to further validate the benefits of clipping in the FA setting. In Figure 4, we observe that replacing the ℓ_1-loss shown in Equation 7 with the Huber or ℓ_2 loss can yield better performance depending on the environment. However, this effect can likely be mitigated by fine-tuning the weight parameter, η.

6 Discussion

In this work, we have given a theoretical foundation for deriving double-sided bounds in reinforcement learning, showcasing their experimental validity. Our investigation in tabular domains has demonstrated that application of these bounds significantly boosts training speed. Coupling our bounds with proof techniques in e.g. (Tang, 2020) may allow for a proof of a faster convergence rate. Beyond the theoretical contributions, our work calls for exploration in several new research directions. While our derived bounds hold in general, there is potential for further refinement given specific classes of value function estimates and transition dynamics or reward function structures, as discussed in Section 3. There is also the potential in transfer learning to leverage bound violation minimization at a state-action level to construct refined initializations from a diverse set of policies.

Integrating our results with other state-of-the-art methods in value-based learning seems a promising direction for future study. Several specific examples include: exploiting ensembles and extending to continuous actor-critic methods, adopting a dynamic schedule for the soft clipping weight parameter, akin to that in (Haarnoja et al., 2018b), and interfacing with model-based approaches such as DYNA (Sutton, 1990), where our tabular results suggest that a more significant performance boost may be achieved. We believe that integrating these methods is an important step to ensuring utility in more complex environments. Finally, we note an intriguing suggestion arising from our experiments, which can be loosely summarized as "clipping is all you need". Further translating the benefits of clipping from tabular to deep RL presents an exciting opportunity for future research.

Acknowledgments

JA would like to acknowledge the use of the supercomputing facilities managed by the Research Computing Department at UMass Boston. The work of JA was supported in part by the College of Science and Mathematics Dean's Doctoral Research Fellowship through support from Oracle, project ID R20000000025727; and in part by the Alliance Innovation Lab – Silicon Valley. RVK and JA would like to acknowledge funding support from the NSF through Award No. DMS-1854350 and PHY-2425180. ST and VM were supported in part by the NSF Award No. 2246221, the Pazy Foundation grant No. 195-2020, and the Alliance Innovation Lab – Silicon Valley.

References

Jacob Adamczyk, Argenis Arriojas, Stas Tiomkin, and Rahul V Kulkarni. Utilizing prior solutions for reward shaping and composition in entropy-regularized reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6658–6665, 2023a.

Jacob Adamczyk, Volodymyr Makarenko, Argenis Arriojas, Stas Tiomkin, and Rahul V. Kulkarni. Bounding the optimal value function in compositional reinforcement learning. In *Proceedings of the Thirty-Ninth Conference on Uncertainty in Artificial Intelligence*, volume 216 of *Proceedings of Machine Learning Research*, pp. 22–32. PMLR, 31 Jul–04 Aug 2023b.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Reincarnating reinforcement learning: Reusing prior computation to accelerate progress. *Advances in Neural Information Processing Systems*, 35:28955–28971, 2022.

Safa Alver and Doina Precup. Constructing a good behavior basis for transfer using generalized policy updates. In *International Conference on Learning Representations*, 2021.

Kavosh Asadi and Michael L Littman. An alternative softmax operator for reinforcement learning. In *International Conference on Machine Learning*, pp. 243–252. PMLR, 2017.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47:235–256, 2002.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym. *arXiv preprint arXiv:1606.01540*, 2016.

Haoyang Cao, Samuel Cohen, and Łukasz Szpruch. Identifiability in inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 34:12362–12373, 2021.

Benjamin Eysenbach and Sergey Levine. Maximum entropy RL (provably) solves some robust RL problems. In *International Conference on Learning Representations*, 2022.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *International Conference on Learning Representations*, 2019.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Tuomas Haarnoja, Vitchyr Pong, Aurick Zhou, Murtaza Dalal, Pieter Abbeel, and Sergey Levine. Composable deep reinforcement learning for robotic manipulation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pp. 6244–6251. IEEE, 2018a.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018b.

Danijar Hafner, Kuang-Huei Lee, Ian Fischer, and Pieter Abbeel. Deep hierarchical planning from pixels. *Advances in Neural Information Processing Systems*, 35:26091–26104, 2022.

Frank S He, Yang Liu, Alexander G Schwing, and Jian Peng. Learning to play in a day: Faster deep reinforcement learning by optimality tightening. In *International Conference on Learning Representations*, 2017.

Sabrina Hoppe and Marc Toussaint. Qgraph-bounded Q-learning: Stabilizing model-free off-policy deep reinforcement learning. *arXiv preprint arXiv:2007.07582*, 2020.

Jaekyeom Kim, Seohong Park, and Gunhee Kim. Constrained GPI for zero-shot transfer in reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4585–4597. Curran Associates, Inc., 2022.

Kimin Lee, Michael Laskin, Aravind Srinivas, and Pieter Abbeel. SUNRISE: A simple unified framework for ensemble learning in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 6131–6141. PMLR, 2021.

Cédric Malherbe and Nicolas Vayatis. Global optimization of Lipschitz functions. In *International Conference on Machine Learning*, pp. 2314–2323. PMLR, 2017.

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):529–533, 2015.

Mark Nemecek and Ronald Parr. Policy caches with successor features. In *International Conference on Machine Learning*, pp. 8025–8033. PMLR, 2021.

Emmanuel Rachelson and Michail G. Lagoudakis. On the locality of action domination in sequential decision making. In *11th International Symposium on Artificial Intelligence and Mathematics (ISIAM 2010)*, pp. 1–8, Fort Lauderdale, US, 2010.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pp. 216–224. Elsevier, 1990.

Yunhao Tang. Self-imitation learning via generalized lower bound Q-learning. *Advances in neural information processing systems*, 33:13964–13975, 2020.

Geraud Nangue Tasse, Steven James, and Benjamin Rosman. Generalisation in lifelong reinforcement learning through logical composition. In *International Conference on Learning Representations*, 2021.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy.* PhD Dissertation, Carnegie Mellon University, 2010.

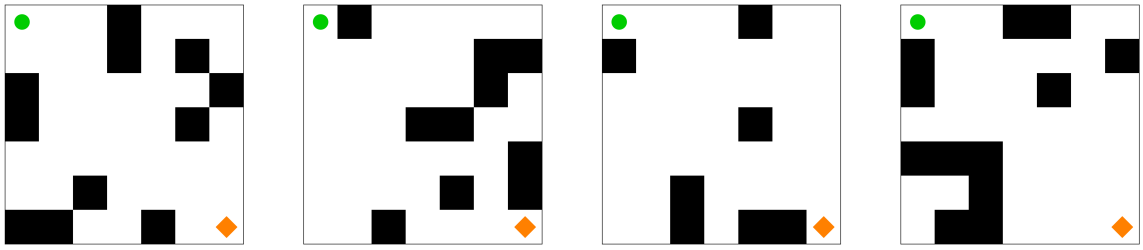

Figure 5: Examples of the random maps generated for the tabular experiments.

A Experiments

In tabular soft Q-learning, we calculate the Bellman residual, mixing it into the current estimate of $Q(s, a)$ at every step taken by the agent. At each update step, we calculate the bounds given by Theorem 1, which are exact in the case that the model is given (red circles in Figure 3). Since these bounds are exact, we can repeatedly take the tightest possible bounds at every step, leading to the consistent fast convergence. When the model is learned through sampling (updating the deterministic reward table and using count-based estimates for the stochastic transition dynamics), the bounds are inexact, so we only use the current step's estimate without iteratively tightening them, which we found to lead to collapse to incorrect values. The clipping performed follows Equation 6 in the main text.

For the tabular experiments, we first generated 30 random mazes for each method to solve 10 times. In each generated 7×7 maze, walls are randomly generated at a site with probability 20%, and a goal is randomly placed at a site without a wall. We show four examples of such mazes in Figure 5. Depth-first search is used to ensure the generated maze has a valid solution (i.e., the rewarding state can be visited by the agent). Each step costs the agent -1, and the goal state incurs a cost of -0.25. The environment is stochastic, such that the probability of moving in the "intended" direction is 75%, and the probability of moving perpendicular to the intended direction is 12.5%. The agent then transitions to the grid state one unit in that direction, as common in MiniGrid or FrozenLake environments. Although we have sparse rewards for the simplicity of environment generation, we find our results to hold across various dense reward settings, with varying levels of stochasticity.

As mentioned in the main text, the ability for bounds to be applied (and training efficiency overall) is sensitive to the scale of the initialized Q-function. We found that a random uniform initialization of the Q-values in the range $(-1, 1)$ performs best for the baseline, and thus we maintain this initialization across all experiments.

Since each maze potentially has a different evaluation reward scale, we normalize the evaluation score so they may be averaged across mazes, akin to e.g. Mnih et al. (2015):

$$\text{Normalized Evaluation Reward} = \frac{\langle R \rangle_{\text{optimal}} - \langle R \rangle_{\text{agent}}}{\langle R \rangle_{\text{agent}} - \langle R \rangle_{\text{uniform}}}, \tag{8}$$

where $\langle R \rangle_{\text{optimal}}$, $\langle R \rangle_{\text{agent}}$, $\langle R \rangle_{\text{uniform}}$ denote the average reward (over 3 episodes) obtained by an agent executing an optimal, training, or uniform random policy, respectively. Finally, we integrate the area under this "Normalized Evaluation Reward" vs. Environment Steps curve, to obtain a metric for the speed of convergence, plotted in Figure 3: "Average Integrated Evaluation Reward (AUC)".

To explore the utility of clipping in function approximator (FA) systems, we use a soft Q-learning (SQL) algorithm Haarnoja et al. (2017), while applying and monitoring clipping given by the bounds in Theorem 1. In particular, we continuously bootstrap by using the previous estimate of the Q-function to generate the bounds, and we clip the target network's output value accordingly. More specifically, we extract bounds from both the target network and Q-network at each step, and take the tighter of the two bounds. For continuous spaces, we use the estimate $\sup r(s, a) \approx \max_{i \in \mathcal{D}} r(s, a)$,

where the max is taken over the current batch (and similarly for $\inf r(s,a)$). We consider the two clipping methods described in the Experiments section of the main text. We have also experimented with different loss functions, as the optimal choice seems to be environment-dependent.

A.1 Implementation Details

For many environments of interest, the transition dynamics are reducible (they have absorbing states returning a termination signal ("done" in Gym Brockman et al. (2016), "terminated" in the newer Gymnasium package Towers et al. (2023)). A common method to assign a Q-value to such states is given by (see e.g. Mnih et al. (2015)):

$$Q(s,a) = \begin{cases} r(s,a) & \text{if terminated} \\ r(s,a) + \gamma V(s') & \text{else} \end{cases} \tag{9}$$

This value assignment means that states near absorbing states will have values $\sim \mathcal{O}(1)$ rather than the $\mathcal{O}(\frac{1}{1-\gamma})$ given by the bounds presented. In passing, we note that one way to circumvent this would be to alter the convention by assigning a value of $r(s,a)/(1-\gamma)$ at termination, since for irreducible dynamics, this would correspond to accumulating the terminal state's reward *ad infinitum*.

To conform to the convention shown above, we modify our bounds to allow for the termination signal to properly affect the bounds. Focusing on deterministic dynamics for simplicity, the bounds are modified from:

$$Q^*(s,a) \leq r(s,a) + \gamma\left(\hat{V}(s') + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a)}{1-\gamma}\right)$$
$$Q^*(s,a) \geq r(s,a) + \gamma\left(\hat{V}(s') + \frac{\min_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a)}{1-\gamma}\right)$$

to the following:

$$Q^*(s,a) \leq r(s,a) + \gamma\left(\hat{V}(s') + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a)}{1-\gamma}\right)[1-\text{done}(\text{s}')]$$
$$Q^*(s,a) \geq r(s,a) + \gamma\left(\hat{V}(s') + \frac{\min_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a)}{1-\gamma}\right)[1-\text{done}(\text{s}')]$$

which can be justified by referring to the value definition used (Equation 9).

Next we provide the algorithm used for clipping in our experiments. We highlight in blue the changes from a standard soft Q-learning approach without clipping, e.g. a discrete-action analogue of Haarnoja et al. (2017).

Algorithm 1 Soft Q-learning with Constant Clipping (Tabular)

Initialize Q-values: $Q(s,a) \sim \text{Unif}(-1,1)$, max sample budget.
Initialize $\text{L}(s,a) = -\infty$, $\text{U}(s,a) = +\infty$.
Set learning rate α, discount factor γ, and inverse temperature β.
while total environment steps $<$ max sample budget **do**
    Reset environment
    while not end of episode **do**
        Choose action $a \sim \pi(\cdot|s) \propto \exp \beta Q(s,a)$
        Take action a: observe reward r, next state s', and termination signal
        Compute state value function: $V(s') = \beta^{-1} \log \mathbb{E}_{a' \sim \pi_0} \exp \beta Q(s',a')$
        Compute the TD error: $\delta = r + \gamma \cdot (1 - \text{terminated}) \cdot V(s') - Q(s,a)$
        Update Q-table: $Q'(s,a) = Q(s,a) + \alpha\delta$
        Calculate new bounds $\{\text{L}'(s,a),\ \text{U}'(s,a)\}$ using Q' in Theorem 1.
        Tighten lower bounds: $\text{L}'(s,a) = \max\{\text{L}'(s,a),\ \text{L}(s,a)\}$
        Tighten upper bounds: $\text{U}'(s,a) = \min\{\text{U}'(s,a),\ \text{U}(s,a)\}$
        Clip the Q-values: $Q'(s,a) = \text{clamp}\left(Q'(s,a), \min = \text{L}'(s,a), \max = \text{U}'(s,a)\right)$
        Update state: $s \leftarrow s'$
        Update Q: $Q \leftarrow Q'$
    end while
end while

Algorithm 2 Soft Q-learning with Conditional TD-updates (Tabular)

Initialize Q-values: $Q(s,a) \sim \text{Unif}(-1,1)$, max sample budget.
Initialize $\text{L}(s,a) = -\infty$, $\text{U}(s,a) = +\infty$.
Set learning rate α, discount factor γ, and inverse temperature β.
while total environment steps $<$ max sample budget **do**
    Reset environment
    while not end of episode **do**
        Choose action $a \sim \pi(\cdot|s) \propto \exp \beta Q(s,a)$
        Take action a: observe reward r, next state s', and termination signal
        Compute state value function: $V(s') = \beta^{-1} \log \mathbb{E}_{a' \sim \pi_0} \exp \beta Q(s',a')$
        Calculate new bounds $\{\text{L}'(s,a),\ \text{U}'(s,a)\}$ using Q' in Equation 2.
        Tighten lower bounds: $\text{L}'(s,a) = \max\{\text{L}'(s,a),\ \text{L}(s,a)\}$
        Tighten upper bounds: $\text{U}'(s,a) = \min\{\text{U}'(s,a),\ \text{U}(s,a)\}$
        Clip the Q-values: $Q'(s,a) = \text{clamp}\left(Q(s,a), \min = \text{L}'(s,a), \max = \text{U}'(s,a)\right)$
        if $Q' == Q$ **then**
            // No clipping has been applied, resort to TD-update:
            Compute the TD error: $\delta = r + \gamma \cdot (1 - \text{terminated}) \cdot V(s') - Q(s,a)$
            Update Q-table: $Q'(s,a) \leftarrow Q'(s,a) + \alpha\delta$
        end if
        Update state: $s \leftarrow s'$
        Update Q: $Q \leftarrow Q'$
        end while
    end while

In Figure 6, we compare Alg. 1 and Alg. 2. Importantly, we find it is imperative to not *always* update the Q-function with TD updates. Rather, we find that by using TD updates only when the Q-values are not changed by clipping, the performance significantly improves in the high learning rate regime.

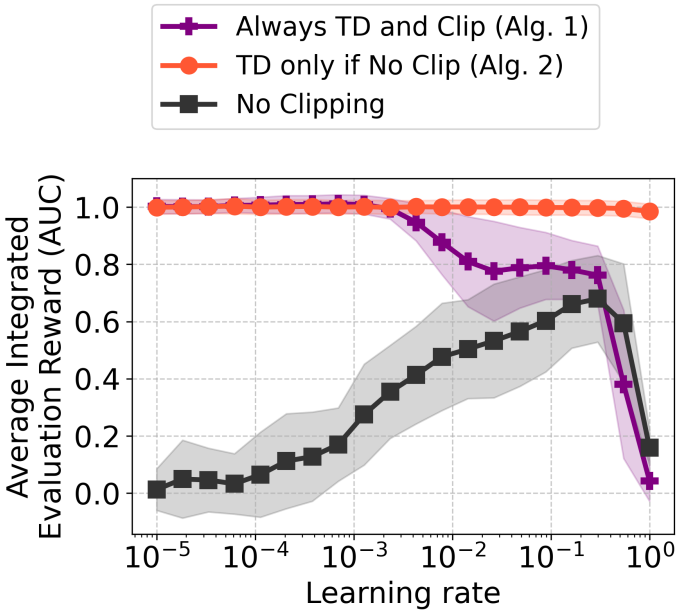

Figure 6: In the 7×7 mazes, we compare the "always clip" (Algorithm 1) and "TD only if no clipping" (Algorithm 2) algorithms as discussed in Section 5. The points labeled "TD only if No Clip" represent the same algorithm shown in the main text's Figure 3, titled "Clipping: Given Model".

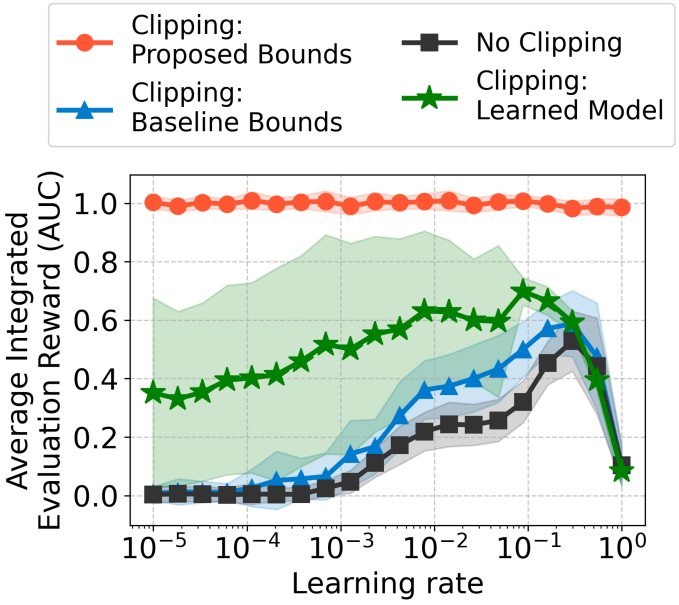

Figure 7: We perform the same experiments as demonstrated in Figure 3, on larger 30×30 mazes, with the same qualitative results.

A.2 Hyperparameters

For the FA classic control experiments, we parameterize the Q-function by an MLP with a standard fixed depth (two hidden layers) and fine-tuned width. We keep the discount factor $\gamma = 0.99$ fixed across tasks, and use a single online and target function. We tune the learning rate, β, target update frequency, training frequency, number of gradient steps per training step, and batch size. The ranges for each hyperparameter, swept uniformly at random, are given below:

Table 1: Hyperparameters and Ranges

Hyperparameter	Range	Sampling Distribution
Learning Rate	$(10^{-4}, 10^{-1})$	Log Uniform
Inverse Temperature, β	$(10^{-2}, 10^{1})$	Log Uniform
Target Update Frequency	$\{1, 10, 100, 1000\}$	Uniform
Training Frequency	$(1, 100)$	Log Uniform
Gradient Steps per Training Step	$(1, 100)$	Log Uniform
Batch Size	$\{2^4, 2^5, 2^6, 2^7, 2^8, 2^9, 2^{10}\}$	Uniform
Hidden Dimension	$\{2^4, 2^5, 2^6, 2^7, 2^8, 2^9\}$	Uniform

We sweep each hyperparameter at random in the ranges shown, and select the best hyperparameter set, sorted by the largest area under the evaluation reward curve (averaged over 3 independent runs). The best hyperparameters for each environment are shown in the next table.

Table 2: fine-tuned Hyperparameters for No Clipping (Baseline) Soft Q-Learning

Environment	CartPole-v1	Acrobot-v1	MountainCar-v0
Learning Rate	0.016	0.0005	0.007
Inverse Temperature, β	0.019	4.5	5.3
Target Update Frequency	1	10	10
Training Frequency	2	2	58
Gradient Steps per Training Step	16	20	5
Batch Size	512	64	128
Learning Starts	0	0	10,000
Hidden Dimension	64	64	512

B Proofs of Exact Results

In this section and the next we provide proofs of the theoretical results in the main text. Each proof is prefaced with a restatement of the theorem for convenience.

We begin with a helpful lemma which bounds the optimal action-value function $Q^*(s,a)$ for any task. We note that these bounds hold for both un-regularized RL and entropy-regularized RL.

Lemma A. *For a task with reward function $r(s,a)$, discount factor γ, the (soft) optimal action-value function $Q^*(s,a)$ satisfies:*

$$Q^*(s,a) \geq r(s,a) + \gamma \frac{\inf_{s,a} r(s,a)}{1-\gamma} \tag{10}$$

$$Q^*(s,a) \leq r(s,a) + \gamma \frac{\sup_{s,a} r(s,a)}{1-\gamma} \tag{11}$$

Proof. We will prove the upper bound here, with the lower bound's proof following similarly. The proof follows from induction on steps (n) of the recursive Bellman backup equation:

$$Q^{(n+1)}(s,a) = r(s,a) + \frac{\gamma}{\beta} \mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)} \log \mathop{\mathbb{E}}_{a'\sim\pi_0(\cdot|s')} \exp\left(\beta Q^{(n)}(s',a')\right). \tag{12}$$

We first use induction to prove

$$Q^{(n)}(s,a) \leq r(s,a) + \gamma \frac{1-\gamma^n}{1-\gamma} \sup_{s,a} r(s,a).$$

Then, since $\lim_{n\to\infty} Q^{(n)}(s,a) = Q^*(s,a)$ and $\gamma \in [0,1)$ the desired result (Equation 11) will follow from this limit.

We set $Q^{(0)}(s,a) = r(s,a)$. The base case ($n=1$) trivially holds:

$$\begin{aligned}
Q^{(1)}(s,a) &= r(s,a) + \frac{\gamma}{\beta} \mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)} \log \mathop{\mathbb{E}}_{a'\sim\pi_0(\cdot|s')} \exp\left(\beta Q^{(0)}(s',a')\right) \\
&= r(s,a) + \frac{\gamma}{\beta} \mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)} \log \mathop{\mathbb{E}}_{a'\sim\pi_0(\cdot|s')} \exp\left(\beta r(s',a')\right) \\
&\leq r(s,a) + \frac{\gamma}{\beta} \sup_{s,a} \left(\beta r(s,a)\right) \\
&= r(s,a) + \gamma \frac{1-\gamma^1}{1-\gamma} \sup_{s,a} r(s,a).
\end{aligned}$$

We proceed in proving the upper bound based on induction as described above. For notational convenience we denote $R \doteq \sup r(s,a)$. The inductive hypothesis is:

$$Q^{(n)}(s,a) \leq r(s,a) + \gamma \frac{1-\gamma^n}{1-\gamma} R. \tag{13}$$

To prove that the inequality holds for step $(n+1)$, we use the Bellman backup equation:

$$\begin{aligned}
Q^{(n+1)}(s,a) &= r(s,a) + \frac{\gamma}{\beta} \mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)} \log \mathop{\mathbb{E}}_{a'\sim\pi_0(\cdot|s')} \exp\left(\beta Q^{(n)}(s',a')\right) \\
Q^{(n+1)}(s,a) &\leq r(s,a) + \frac{\gamma}{\beta} \mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)} \log \mathop{\mathbb{E}}_{a'\sim\pi_0(\cdot|s')} \exp\left(\beta \left[r(s',a') + \gamma \frac{1-\gamma^n}{1-\gamma} R\right]\right) \\
&\leq r(s,a) + \gamma \left(R + \gamma \frac{1-\gamma^n}{1-\gamma} R\right)
\end{aligned}$$

At this point if the transition dynamics were known then one could improve this bound by including the next term, $\mathbb{E}_{s'\sim p(\cdot|s,a)} \max_{a'} r(s', a')$. Instead we do not assume access to the next term, bounding this term by R. Then we have:

$$\begin{aligned} Q^{(n+1)}(s,a) &\leq r(s,a) + \gamma \left(R + \gamma \frac{1-\gamma^n}{1-\gamma} R \right) \\ &= r(s,a) + \gamma \frac{1-\gamma^{n+1}}{1-\gamma} R, \end{aligned}$$

which completes the proof of the inductive step. As stated above, this completes the proof of the upper bound by taking the limit $n \to \infty$. The lower bound follows similarly by swapping all inequalities and taking inf instead of sup. □

We now proceed with the proof of our first result, Theorem 1. We do so by applying Lemma A to the K^* function of Adamczyk et al. (2023a)'s Theorem 1.

Theorem 1. *Consider an entropy-regularized MDP $\langle \mathcal{S}, \mathcal{A}, p, r, \gamma, \beta, \pi_0 \rangle$ with optimal value function $Q^*(s,a)$. Let any bounded function $Q(s,a)$ be given. Denote the corresponding state-value function as $V(s) \doteq 1/\beta \log \mathbb{E}_{a\sim\pi_0} \exp \beta Q(s,a)$. Then, $Q^*(s,a)$ is bounded by:*

$$r(s,a) + \gamma \left(\mathop{\mathbb{E}}_{s'\sim p} V(s') + \frac{\inf \Delta}{1-\gamma} \right) \leq Q^*(s,a) \leq r(s,a) + \gamma \left(\mathop{\mathbb{E}}_{s'\sim p} V(s') + \frac{\sup \Delta}{1-\gamma} \right) \tag{14}$$

where

$$\Delta(s,a) \doteq r(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim p} V(s') - Q(s,a).$$

Proof. As a point of notation, $\widetilde{r}(s,a)$ in Adamczyk et al. (2023a) is the same as our $r(s,a)$. Their $r(s,a)$ is now replaced by the reward function corresponding to an "optimal" value function of $Q(s,a)$. As discussed, $Q(s,a)$ need not be an optimal value function corresponding to any known or desired task (reward function). However, because of Theorem 1 in Cao et al. (2021), we see that choosing a reward function of $Q(s,a) - \gamma \mathbb{E}_{s'} V(s')$ ensures that $Q(s,a)$ is indeed an *optimal* value function, allowing us to apply Theorem 1 of Adamczyk et al. (2023a):

$$Q^*(s,a) = Q(s,a) + K^*(s,a) \tag{15}$$

where K^* is the optimal soft action value function corresponding to a task with reward function $\Delta(s,a) \doteq r(s,a) + \gamma \mathbb{E}_{s'\sim p(\cdot|s,a)} V(s') - Q(s,a)$. By applying Lemma A to the value function K^*, we arrive at the stated result in Equation 14:

$$\begin{aligned} Q^*(s,a) &= Q(s,a) + K^*(s,a) \\ &\leq Q(s,a) + \Delta(s,a) + \gamma \frac{\sup \Delta}{1-\gamma} \\ &= Q(s,a) + r(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)} V(s') - Q(s,a) + \gamma \frac{\sup \Delta}{1-\gamma} \\ &= r(s,a) + \gamma \left(\mathop{\mathbb{E}}_{s'\sim p(\cdot|s,a)} V(s') + \frac{\sup \Delta}{1-\gamma} \right). \end{aligned}$$

The same proof holds for the lower bound. □

We now turn to the proof of the convergence result presented in the main text:

Proposition 1. *Let $B(\cdot)$ denote the Bellman operator, and let the functions $L(s,a)$, $U(s,a)$ be lower and upper bounds on the optimal action-value function respectively: $L(s,a) \leq Q^*(s,a) \leq U(s,a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The clipped Bellman operator, $B_C Q(s,a) := \max\left(\min\left(BQ(s,a), U(s,a)\right), L(s,a)\right)$ converges to the optimal action-value function $Q^*(s,a) = B^\infty Q(s,a) = B_C^\infty Q(s,a)$ for any bounded initial function $Q(s,a)$.*

Proof. We first show convergence of the operator B_C, then show that it converges to the same fixed point as that of B. For convergence, it suffices to show that B_C is a contraction mapping, that is: $|B_C Q(s,a) - Q^*(s,a)| \leq \gamma \, |Q(s,a) - Q^*(s,a)|$.

There are three cases for the magnitude of $BQ(s,a)$ relative to the upper and lower bounds:

1. $BQ(s,a) \in (L(s,a), U(s,a))$
2. $BQ(s,a) \in (-\infty, L(s,a))$
3. $BQ(s,a) \in (U(s,a), \infty)$

In the first case, clipping does not occur and hence $B_C Q(s,a) = BQ(s,a)$, which contracts with rate γ. In the second case, we can write $BQ(s,a) = L(s,a) - \chi(s,a)$ where $\chi(s,a) := BQ(s,a) - L(s,a) > 0$ is referred to as the "bound violation". Then,

$$\begin{aligned}
|B_C Q(s,a) - Q^*(s,a)| &= |Q^*(s,a) - B_C Q(s,a)| \\
&= |Q^*(s,a) - L(s,a)| \\
&\leq |Q^*(s,a) - L(s,a) + \chi(s,a)| \\
&= |Q^*(s,a) - (L(s,a) - \chi(s,a))| \\
&= |Q^*(s,a) - BQ(s,a)| \\
&\leq \gamma |Q(s,a) - Q^*(s,a)|.
\end{aligned}$$

A similar proof holds for the third case.

By the Banach fixed point theorem, it follows that repeated application of B_C converges to a fixed point. It is clear that the fixed point for B is also a fixed point for B_C, and since it is unique, we have $B_C^\infty Q(s,a) = B^\infty Q(s,a) = Q^*(s,a)$. □

C Error Analysis for Continuous Spaces

In this section, we turn to those results specific to the bounds in continuous spaces and their error analysis, based on Lipschitz continuity and finite sampling errors.

C.1 Reward functions

In theoretical analyses of RL algorithms it is typical to assume a bounded reward function: $r(s,a) \in (R_{\min}, R_{\max})$ for all $s \in \mathcal{S}, a \in \mathcal{A}$. However, the values of these bounds may *not* be known to the agent (or even RL practitioner) in the general model-free case. Thus, one must resort to sampling the reward and estimate the values of $R_{\min}$ and $R_{\max}$. In fact, due to Corollary 1 from Malherbe & Vayatis (2017) one can obtain global empirical bounds on $r(s,a)$ with high probability. In the following, we restate Corollary 1 for convenience. Notice that the resulting bounds depend only on a dataset (replay buffer or batch) $\mathcal{B}$, the dimensionality of the continuous state-action space $|\mathcal{S} \times \mathcal{A}|$, a desired probability $1 - \delta$, and confidence interval ε. Given these input parameters, the

global extrema of the reward function (or any Lipschitz function) can be bounded with high confidence, within the convex hull of points sampled (i.e. the replay buffer). For bounds on the reward function, one obtains the following:

Corollary 1. *Consider an MDP with bounded continuous state-action space $\mathcal{S} \times \mathcal{A} \subset \mathbb{R}^d$, and deterministic dynamics. Let $\varepsilon > 0$, $\delta > 0$, and $|\mathcal{B}| \geq \left(\frac{L_r \mathrm{diam}(\mathcal{S}\times\mathcal{A})}{\varepsilon}\right)^d \log 1/\delta$, be given, where* diam *represents the diameter of a bounded space. Suppose $|\mathcal{B}|$ samples are drawn uniformly from state-action space, $(s,a) \sim \mathrm{Unif}\,(\mathcal{S} \times \mathcal{A})$, and denote the convex hull of these points as $c \doteq \mathrm{Conv}(\mathcal{B})$. Then, the following bounds on the reward function's extrema*

$$\inf_c r(s,a) \geq \min_{(s,a)\in\mathcal{B}} r(s,a) - \varepsilon,$$
$$\sup_c r(s,a) \leq \max_{(s,a)\in\mathcal{B}} r(s,a) + \varepsilon,$$

hold with probability at least $1-\delta$.

Proof. The only difference from the result in Malherbe & Vayatis (2017) is that we have written the result in terms of the number of samples, which is found by solving for $|\mathcal{B}|$:

$$\varepsilon \leq L_r \cdot \mathrm{diam}(c) \cdot \left(\frac{\log(1/\delta)}{|\mathcal{B}|}\right)^{\frac{1}{d}} \tag{16}$$

$$|\mathcal{B}| \geq \left(\frac{L_r \mathrm{diam}(c)}{\varepsilon}\right)^d \log 1/\delta. \tag{17}$$

Note that these bounds only hold within the convex hull of the sampled points. □

C.2 Lipschitz Continuity

Due to the hypotheses of Corollary 1, the function of interest must be Lipschitz continuous. In the present case, this function (the one being maximized or minimized) is Δ, as seen in Theorem 1. Therefore we must derive the Lipschitz constant for the function Δ. To carry out this calculation, we suppose that a Lipschitz MDP and Lipschitz input function, denoted $Q(s,a)$, are given.

Lemma B. *Consider an MDP with (L_r, L_p)-Lipschitz rewards and dynamics, L_κ-Lipschitz continuous $\log \pi_0(\cdot|s)$ (with respect to s) and L_Q-Lipschitz continuous function $Q(s,a)$. The function Δ in Equation 1 generated for this MDP with Q is Lipschitz-continuous with constant*

$$L_\Delta = L_r + L_Q + \gamma L_p (L_Q + \beta^{-1} L_\kappa).$$

In the case of a uniform prior policy π_0 this simplifies to

$$L_\Delta = L_r + (1 + \gamma L_p) L_Q. \tag{18}$$

Proof. The sum of Lipschitz functions is itself Lipschitz continuous, with the Lipschitz constant being the sum of all terms' Lipschitz constants (through the triangle inequality). We begin with the calculation of the Lipschitz constant for the soft state-value function, V. First, we note that the operation of LogSumExp is Lipschitz continuous with Lipschitz constant 1 (cf. "mellowmax" with an additive constant in Asadi & Littman (2017)). Since this operation (over action space) is

composed with the L_Q-Lipschitz input Q-function and L_κ-Lipschitz prior $\log \pi_0$ we have for $V(s')$,

$$\beta^{-1} \log \mathop{\mathbb{E}}_{a' \sim \pi_0(\cdot|s')} \exp \beta Q(s', a') = \beta^{-1} \log \sum_a \exp \beta \left(Q(s', a') + \beta^{-1} \log \pi_0(a'|s') \right). \tag{19}$$

Written this way, we can see the operation in question is a composition and sum of Lipschitz functions, leading to a Lipschitz constant of $L_V = 1 \cdot (L_Q + \beta^{-1} L_\kappa)$. Note that in the case of a uniform prior policy $L_\kappa = 0$, and the Lipschitz constant for the state value function reduces to L_Q.

Now calculating the Lipschitz constant of Δ, the contribution from $\mathbb{E}_{s' \sim p} V(s')$ is:

$$\left| \mathop{\mathbb{E}}_{s' \sim p(\cdot|s,a)} V(s') - \mathop{\mathbb{E}}_{s' \sim p(\cdot|\hat{s},\hat{a})} V(s') \right| = \left| \int_{\mathcal{S}} \left(p(\cdot|s,a) - p(\cdot|\hat{s},\hat{a}) \right) V(s') ds' \right| \tag{20}$$

$$\leq L_p L_V \left(|s - \hat{s}| + |a - \hat{a}| \right). \tag{21}$$

In the second line we have used the same argument as in the proof of Lemma 2 of Rachelson & Lagoudakis (2010). Now, using the full definition of Δ we may finally compute its Lipschitz constant as:

$$\begin{aligned}
|\Delta(s,a) - \Delta(\hat{s},\hat{a})| &= \left| r(s,a) + \gamma \mathop{\mathbb{E}}_{s' \sim p(\cdot|s,a)} V(s') - Q(s,a) - \left(r(\hat{s},\hat{a}) + \gamma \mathop{\mathbb{E}}_{s' \sim p(\cdot|\hat{s},\hat{a})} V(s') - Q(\hat{s},\hat{a}) \right) \right| \\
&\leq (L_r + L_Q) \left(|s - \hat{s}| + |a - \hat{a}| \right) + \gamma \left| \mathop{\mathbb{E}}_{s' \sim p(\cdot|s,a)} V(s') - \mathop{\mathbb{E}}_{s' \sim p(\cdot|\hat{s},\hat{a})} V(s') \right| \\
&\leq (L_r + L_Q) \left(|s - \hat{s}| + |a - \hat{a}| \right) + \gamma L_p L_V \left(|s - \hat{s}| + |a - \hat{a}| \right) \\
&= \left(L_r + L_Q + \gamma L_p (L_Q + \beta^{-1} L_\kappa) \right) \left(|s - \hat{s}| + |a - \hat{a}| \right).
\end{aligned}$$

The second line follows from the triangle inequality and Lipschitz continuity of the reward function and input function Q. This allows us to read off the final Lipschitz constant as:

$$L_\Delta = L_r + L_Q + \gamma L_p (L_Q + \beta^{-1} L_\kappa). \tag{22}$$

Note that in the simpler case of the MaxEnt uniform prior policy ($L_\kappa = 0$) the Lipschitz constant of Δ simplifies to $L_\Delta = L_r + (1 + \gamma L_p) L_Q$. □

C.3 Extension of Exact Bounds

In this section, we extend our results from the tabular case (Theorem 1) to scenarios where sampling is required (i.e. in the presence of continuous state-action spaces and stochastic dynamics).

We will proceed by introducing and proving three progressively more involved results, covering the following situations: (1) Sampling error arises from estimating the extrema of Δ, which can be calculated exactly, but for which we do not have access to global extrema. (2) An additional sampling error arises due to stochastic transition dynamics. (3) Additional sampling error arises due to continuous action spaces, for which the state-value function integral cannot be calculated exactly.

In each case, we provide the number of samples required for a given (ε, δ)-concentration inequality. In (1) we denote the number of samples for estimating the extrema of Δ with $|\mathcal{B}|$ samples (as in practice we sample using the current batch of replay data), and in (2) we introduce $n_{\mathcal{S}}$, the number of samples for the next-state transitions, and in (3) we introduce $n_{\mathcal{A}}$, the number of samples for next-actions drawn from the prior policy.

Note that for partially discrete spaces (e.g. continuous state, discrete action), we assume that optimization over the discrete variable is feasible. As in Corollary 1, all proceeding bounds involving extremization over Δ only hold within the convex hull of the sampled points.

Theorem 1. *Consider an MDP with bounded continuous state space, discrete actions, and determinstic dynamics. Let* $\varepsilon_1 > 0$, $\delta_1 > 0$, L_Δ *given in Lemma B, and* $|\mathcal{B}| \geq \left(\frac{L_\Delta \operatorname{diam}(c)}{\varepsilon_1}\right)^{|\mathcal{S}|} \log 1/\delta_1$, *be given. Suppose* $|\mathcal{B}|$ *samples are drawn uniformly from state-space,* $s \sim \operatorname{Unif}(\mathcal{S})$. *Then, the following bounds on the Q-values*

$$Q^*(s,a) \leq r(s,a) + \gamma \left(\mathop{\mathbb{E}}_{s' \sim p} V(s') + \frac{\max_{(s,a)\in\mathcal{B}} \Delta(s,a) + \varepsilon_1}{1-\gamma} \right) \tag{23}$$

$$Q^*(s,a) \geq r(s,a) + \gamma \left(\mathop{\mathbb{E}}_{s' \sim p} V(s') + \frac{\min_{(s,a)\in\mathcal{B}} \Delta(s,a) - \varepsilon_1}{1-\gamma} \right) \tag{24}$$

$$\tag{25}$$

hold with probability at least $1-\delta_1$, *where* Δ *is given by Equation 1.*

Proof. The bounds follow directly from applying Corollary 1 to the bounds given in Theorem 1. □

For the case of stochastic dynamics, we must construct an estimate of V and Δ which can be calculated with the given information (samples of the next state, rather than an exact integral):

$$\mathop{\mathbb{E}}_{s' \sim p(\cdot|s,a)} V(s') = \int_{\mathcal{S}} p(\cdot|s,a) V(s') ds' \to \frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} V(s'_i) \tag{26}$$

$$\Delta(s,a) \to \hat{\Delta}(s,a) = r(s,a) + \gamma \frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} V(s'_i) - Q(s,a) \tag{27}$$

where $n_{\mathcal{S}}$ denotes the number of next-state samples from the transition dynamics. Based on these definitions, we introduce two small lemmas, bounding the error in replacing the true functions with their corresponding estimates:

Lemma C. *Let* $\delta > 0$ *and* $\varepsilon > 0$ *be given. Then with at least* $n \geq \left(\frac{R_{\max} - R_{\min}}{\varepsilon(1-\gamma)}\right)^2 \log \frac{2}{\delta}$ *samples on the next state, the error in replacing* $\mathbb{E}_{s'} V(s')$ *with* $\frac{1}{n}\sum_{i=1}^n V(s'_i)$ *is bounded with probability* $1-\delta$, *leading to the following bounds:*

$$\mathop{\mathbb{E}}_{s' \sim p} V(s') \leq \frac{1}{n} \sum_{i=1}^{n} V(s'_i) + \varepsilon \tag{28}$$

$$\mathop{\mathbb{E}}_{s' \sim p} V(s') \geq \frac{1}{n} \sum_{i=1}^{n} V(s'_i) - \varepsilon. \tag{29}$$

Proof. Applying Hoeffding's inequality on the relevant term gives

$$\mathbb{P}\left(\left| \mathop{\mathbb{E}}_{s' \sim p} V(s') - \frac{1}{n} \sum_{i=1}^{n} V(s'_i) \right| < \varepsilon \right) \geq 1 - 2\exp\left(-\frac{2\varepsilon^2}{b^2} n \right)$$

where as usual $b = (R_{\max} - R_{\min})(1-\gamma)^{-1}$ is the gap between a lower and upper bound on the concentrating quantity of interest, V. Note that here and in the following we assume exact global bounds on the reward function, $r(s,a)$ though in principal a corresponding error term based on finite

samples can be included. Defining $\delta = 2\exp\left(-\frac{2\varepsilon^2}{b^2}n\right)$, we have with probability at least $1-\delta$:

$$\left|\mathop{\mathbb{E}}_{s'\sim p} V(s') - \frac{1}{n}\sum_{i=1}^{n} V(s'_i)\right| < \varepsilon, \tag{30}$$

where solving for n yields a requirement of $n = \frac{1}{2}\left(\frac{R_{\max}-R_{\min}}{\varepsilon(1-\gamma)}\right)^2 \log\frac{2}{\delta}$ samples. Expanding the absolute values leads to the two bounds shown, which is a more useful form for the subsequent results. □

We next provide a similar lemma for the error in replacing Δ with $\hat{\Delta}$:

Lemma D. *Let $\varepsilon > 0$ and $\delta > 0$ be given. Then given $n \geq \frac{1}{2}\left(\frac{R_{\max}-R_{\min}}{\varepsilon(1-\gamma)}\right)^2 \log\frac{2}{\delta}$ samples to estimate the value of the next state, the error in replacing $\Delta(s,a)$ with $\hat{\Delta}(s,a)$ in Equation 27 is upper bounded. That is, for all $s \in \mathcal{S}, a \in \mathcal{A}$:*

$$\left|\Delta(s,a) - \hat{\Delta}(s,a)\right| \leq \gamma\varepsilon \tag{31}$$

with probability $1-\delta$.

Proof. From the definitions, we can immediately calculate the following error bound

$$\begin{aligned}
\left|\Delta(s,a) - \hat{\Delta}(s,a)\right| &= \left|r(s,a) + \gamma \mathop{\mathbb{E}}_{s'\sim p} V(s') - Q(s,a) - \left(r(s,a) + \gamma\frac{1}{n}\sum_{i=1}^{n} V(s'_i) - Q(s,a)\right)\right| \\
&= \gamma\left|\mathop{\mathbb{E}}_{s'\sim p} V(s') - \frac{1}{n}\sum_{i=1}^{n} V(s'_i)\right| \\
&\leq \gamma\varepsilon.
\end{aligned}$$

where the last line holds with probability $1-\delta$. The last line follows from Lemma C when using at least $n = \frac{1}{2}\left(\frac{R_{\max}-R_{\min}}{\varepsilon(1-\gamma)}\right)^2 \log\frac{2}{\delta}$ samples. □

Now we combine all the previous results to arrive at the following extension of our main results to the case of sampling in stochastic environments with continuous state space:

Theorem 2. *Consider an MDP with bounded continuous state space, discrete actions and stochastic dynamics. Let $\varepsilon_1, \varepsilon_2 > 0$, $\delta_1, \delta_2 > 0$, and $|\mathcal{B}| \geq \left(\frac{L_\Delta \operatorname{diam}(c)}{\varepsilon_1}\right)^{|\mathcal{S}|} \log 1/\delta_1$, $n_{\mathcal{S}} \geq \frac{1}{2}\left(\frac{R_{\max}-R_{\min}}{\varepsilon_2(1-\gamma)}\right)^2 \log\frac{2}{\delta_2}$, be given. Suppose $|\mathcal{B}|$ samples are drawn uniformly from state-space, $s \sim \operatorname{Unif}(\mathcal{S})$. Then, for V, $\hat{\Delta}$ given in Equation 26, 27, the following bounds on the Q-values*

$$Q^*(s,a) \leq r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} V(s'_i) + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \varepsilon_1 + \varepsilon_2}{1-\gamma}\right) \tag{32}$$

$$Q^*(s,a) \geq r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} V(s'_i) + \frac{\min_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) - \varepsilon_1 - \varepsilon_2}{1-\gamma}\right) \tag{33}$$

hold with probability at least $1-\delta_1-2\delta_2$.

Proof. For stochastic dynamics there remains an error in using samples to estimate the expectation over next states in $V(s')$.

Now, for simplicity, we will assume that the same number of samples $n_{\mathcal{S}}$ are used to estimate $V(s')$ appearing explicitly in the bound *and* implicitly in the definition of Δ. In principle, these can be different values, but we propose (as done experimentally) to use the same batch of replay data for both calculations.

Then by Lemma C, let $\varepsilon_2 > 0, \delta_2 > 0$ and $n_{\mathcal{S}} \geq \frac{1}{2}\left(\frac{R_{\max}-R_{\min}}{\varepsilon_2(1-\gamma)}\right)^2 \log\frac{2}{\delta_2}$ next-state samples be given. Then with probability at least $1-\delta_2$:

$$\Delta(s,a) \leq \hat{\Delta}(s,a) + \gamma\varepsilon_2 \tag{34}$$

$$\max_{(s,a)\in\mathcal{B}} \Delta(s,a) \leq \max_{(s,a)\in\mathcal{B}} \left(\hat{\Delta}(s,a) + \gamma\varepsilon_2\right) \tag{35}$$

$$\leq \max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \gamma\varepsilon_2. \tag{36}$$

Combining the bound above with Lemma B allows one to replace all instances of V and Δ with their approximations in the upper bound of Theorem 1:

$$\begin{aligned}
Q^*(s,a) &\leq r(s,a) + \gamma\left(\mathop{\mathbb{E}}_{s'\sim p} V(s') + \frac{\max_{(s,a)\in\mathcal{B}} \Delta(s,a) + \varepsilon_1}{1-\gamma}\right) \\
&\leq r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} V(s') + \varepsilon_2 + \frac{\max_{(s,a)\in\mathcal{B}} \Delta(s,a) + \varepsilon_1}{1-\gamma}\right) \\
&\leq r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} V(s') + \varepsilon_2 + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \gamma\varepsilon_2 + \varepsilon_1}{1-\gamma}\right) \\
&= r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} V(s') + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \varepsilon_1 + \varepsilon_2}{1-\gamma}\right),
\end{aligned}$$

which holds with probability at least $(1-\delta_1)(1-\delta_2)^2 \geq 1-\delta_1-2\delta_2$ (we ignore the terms beyond first order which are negligible in the limit of small δ_i). One factor of $1-\delta_1$ arises from Theorem 1 and two factors of $1-\delta_2$ correspond to the use of Lemma B and Lemma C separately, as they act on independent next-state samples: Lemma B operates on the next-state samples dictated by the fixed (s,a)-value of interest on the left-hand side of the bound, whereas Lemma C operates on the next states in the batch $\mathcal{B}$. As discussed previously, we have assumed for simplicity the same values of $(\varepsilon_2, \delta_2)$ in their corresponding concentration bounds.

A similar proof holds for the lower bound. □

Lastly, we will consider the case of continuous actions in a soft Q-learning style algorithm where one samples actions from the prior policy π_0 to estimate the following integral

$$V(s) = \beta^{-1}\log\int_{\mathcal{A}} e^{\beta Q(s,a)}\pi_0(a|s)da, \tag{37}$$

specifically with $n_{\mathcal{A}}$ samples from the prior policy,

$$\hat{V}(s) \doteq \beta^{-1}\log\sum_{i=1}^{n_{\mathcal{A}}} e^{\beta Q(s,a_i)}. \tag{38}$$

Theorem 3. *Consider an MDP with a bounded continuous state and action space, $\mathcal{S} \times \mathcal{A} \subset \mathbb{R}^d$, with stochastic dynamics. Suppose an L_Q-Lipschitz function $Q(s,a)$ is given to generate double-sided bounds on the optimal value function, denoted $Q^*(s,a)$. Let $\varepsilon_i > 0, \delta_i > 0$ be given and define the horizon $H = (1-\gamma)^{-1}$, and sample budgets*

$$|\mathcal{B}| \geq \left(\frac{L_\Delta \operatorname{diam}(\mathcal{S} \times \mathcal{A})}{\varepsilon_1}\right)^d \log \frac{1}{\delta_1}, \tag{39}$$

$$n_{\mathcal{S}} \geq \frac{1}{2}\left(\frac{H(R_{\max} - R_{\min})}{\varepsilon_2}\right)^2 \log \frac{2}{\delta_2}, \tag{40}$$

$$n_{\mathcal{A}} \geq \frac{1}{2}\left(\frac{e^{\beta H(R_{\max} - R_{\min})} - 1}{e^{\beta \varepsilon_3} - 1}\right)^2 \log \frac{2}{\delta_3}. \tag{41}$$

Suppose $|\mathcal{B}|$ samples are drawn uniformly from the state-action space, $s \sim \operatorname{Unif}(\mathcal{S})$ and $a \sim \operatorname{Unif}(\mathcal{A})$ to estimate the extrema of $\hat{\Delta}$. Suppose $n_{\mathcal{S}}$ samples are used to estimate the expectation over next-states and $n_{\mathcal{A}}$ samples are used to estimate the soft state-value function Denoting $\hat{V}, \hat{\Delta}$ as the quantities estimated from samples, the following bounds on the Q-values

$$Q^*(s,a) \leq r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} \hat{V}(s') + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \varepsilon_1 + \varepsilon_2 + \varepsilon_3}{1-\gamma}\right) \tag{42}$$

$$Q^*(s,a) \geq r(s,a) + \gamma\left(\frac{1}{n_{\mathcal{S}}}\sum_{i=1}^{n_{\mathcal{S}}} \hat{V}(s') + \frac{\min_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) - \varepsilon_1 - \varepsilon_2 - \varepsilon_3}{1-\gamma}\right) \tag{43}$$

hold with probability at least $1 - \delta_1 - 2\delta_2 - 2\delta_3$.

As before, we first provide a bound on the error in replacing V with $\hat{V}$ before combining it with the previous result.

Lemma E. *Let the definitions in Equations 37, 38 and some $\varepsilon > 0,\ \delta > 0$ be given. Then, for*

$$n \geq \frac{1}{2}\left(\frac{e^{\beta(R_{\max} - R_{\min})/(1-\gamma)} - 1}{e^{\beta\varepsilon} - 1}\right)^2 \log \frac{2}{\delta} \tag{44}$$

action samples from the prior policy $a \sim \pi_0(a|s)$, the error in replacing the state value function with its estimate is bounded:

$$\frac{1}{n}\sum_{i=1}^{n} e^{\beta Q(s,a_i)} - \varepsilon < V(s) < \frac{1}{n}\sum_{i=1}^{n} e^{\beta Q(s,a_i)} + \varepsilon \tag{45}$$

with probability $1 - \delta$.

Proof. We will focus on the proof of the upper bound. The lower bound follows from a similar proof. From Hoeffding's inequality, with probability $1 - \delta$,

$$\left| e^{\beta V(s)} - \frac{1}{n}\sum_{i=1}^{n} e^{\beta Q(s,a_i)} \right| < \left(e^{\beta R_{\max}/(1-\gamma)} - e^{\beta R_{\min}/(1-\gamma)}\right)\sqrt{\frac{1}{2n}\log\frac{2}{\delta}} \doteq \widetilde{\varepsilon}$$

$$e^{\beta V(s)} \leq e^{\beta \hat{V}(s)} + \widetilde{\varepsilon}.$$

By taking the log on both sides

$$\begin{aligned} V(s) &\leq \frac{1}{\beta}\log(e^{\beta\hat{V}(s)} + \widetilde{\varepsilon}) \\ &= \hat{V}(s) + \beta^{-1}\log(1 + \widetilde{\varepsilon}e^{-\beta\hat{V}(s)}) \\ &\leq \hat{V}(s) + \beta^{-1}\log\left(1 + \widetilde{\varepsilon}e^{-\beta R_{\min}/(1-\gamma)}\right) \\ &\doteq \hat{V}(s) + \varepsilon_+ \end{aligned}$$

(with probability $1-\delta$), where ε_+ satisfies

$$\frac{e^{\beta\varepsilon_+} - 1}{e^{-\beta R_{\min}/(1-\gamma)}} = \widetilde{\varepsilon} = \left(e^{\beta R_{\max}/(1-\gamma)} - e^{\beta R_{\min}/(1-\gamma)}\right)\sqrt{\frac{1}{2n_+}\log\frac{2}{\delta}}. \tag{46}$$

or in other words,

$$n_+ = \frac{1}{2}\log\frac{2}{\delta}\left(\frac{e^{\beta(R_{\max}-R_{\min})/(1-\gamma)} - 1}{e^{\beta\varepsilon_+} - 1}\right)^2. \tag{47}$$

For clarity we also provide the corresponding lower bound here:

$$\begin{aligned} e^{\beta V(s)} &\geq e^{\beta\hat{V}(s)} - \widetilde{\varepsilon} \\ V(s) &\geq \frac{1}{\beta}\log\left(e^{\beta\hat{V}(s)} - \widetilde{\varepsilon}\right) \\ &= \hat{V}(s) + \beta^{-1}\log(1 - \widetilde{\varepsilon}e^{-\beta\hat{V}(s)}) \\ &\geq \hat{V}(s) + \beta^{-1}\log(1 - \widetilde{\varepsilon}e^{-\beta R_{\max}/(1-\gamma)}) \\ &\doteq \hat{V}(s) - \varepsilon_- \end{aligned}$$

where again we solve for n_-:

$$n_- = \frac{1}{2}\log\frac{2}{\delta}\left(\frac{1 - e^{-\beta(R_{\max}-R_{\min})/(1-\gamma)}}{1 - e^{-\beta\varepsilon_-}}\right)^2. \tag{48}$$

Letting $\varepsilon_+ = \varepsilon_- \doteq \varepsilon$ for simplicity and solving for the value of n in Hoeffding's inequality such that both the upper and lower bounds on V are satisfied gives:

$$n \geq \max\{n_+, n_-\} \tag{49}$$

$$= \frac{1}{2}\log\frac{2}{\delta}\max\left\{\left(\frac{e^{\beta(R_{\max}-R_{\min})/(1-\gamma)} - 1}{e^{\beta\varepsilon} - 1}\right)^2, \left(\frac{1 - e^{-\beta(R_{\max}-R_{\min})/(1-\gamma)}}{1 - e^{-\beta\varepsilon}}\right)^2\right\}. \tag{50}$$

For small $\varepsilon < (R_{\max} - R_{\min})/(1-\gamma)$, the first term dominates, thus

$$n \geq \frac{1}{2}\log\frac{2}{\delta}\left(\frac{e^{\beta(R_{\max}-R_{\min})/(1-\gamma)} - 1}{e^{\beta\varepsilon} - 1}\right)^2 \tag{51}$$

samples suffice to satisfy both the lower and upper bounds $|V - \hat{V}| < \varepsilon$ with probability at least $1-\delta$. In passing, we note that in the low β regime, $\beta\varepsilon \ll 1$ and $\beta(R_{\max} - R_{\min})/(1-\gamma) \ll 1$, the required samples simplifies to the "usual" (Hoeffding) form, quadratic in H/ε:

$$n(\beta \ll 1) \geq \frac{1}{2}\left(\frac{H(R_{\max} - R_{\min})}{\varepsilon}\right)^2 \log\frac{2}{\delta}. \tag{52}$$

□

Now, to prove Theorem 3, we apply the same techniques as before:

Proof. Applying Lemma E to the expectation over actions in $\hat{V}$ and $\hat{\Delta}$ in Theorem 2 gives, similar to the previous proof, two terms of ε_3:

$$Q^*(s,a) \leq r(s,a) + \gamma \left(\frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} V(s') + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \varepsilon_1 + \varepsilon_2}{1-\gamma} \right) \tag{53}$$

$$\leq r(s,a) + \gamma \left(\frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} \hat{V}(s') + \varepsilon_3 + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \varepsilon_1 + \varepsilon_2 + \gamma\varepsilon_3}{1-\gamma} \right) \tag{54}$$

$$= r(s,a) + \gamma \left(\frac{1}{n_{\mathcal{S}}} \sum_{i=1}^{n_{\mathcal{S}}} \hat{V}(s') + \frac{\max_{(s,a)\in\mathcal{B}} \hat{\Delta}(s,a) + \varepsilon_1 + \varepsilon_2 + \varepsilon_3}{1-\gamma} \right). \tag{55}$$

Similar to the proof of Theorem 2, we introduced two instances of this action sampling (one for $V(s')$ and one for the extrema of $\hat{\Delta}(s,a)$). This requires an additional two factors of $1-\delta_3$ in the confidence: $(1-\delta_1)(1-\delta_2)^2(1-\delta_3)^2 \geq 1-\delta_1-2\delta_2-2\delta_3$. □

We note that one can also instead combine Theorem 1 with Lemma E to arrive at double-sided bounds for the case of deterministic transitions with continuous actions.