Journal of Machine Learning Research 25 (2024) 1-48 Submitted 1/24; Revised 6/24; Published 8/24

Rethinking Discount Regularization: New Interpretations, Unintended Consequences, and Solutions for Regularization in Reinforcement Learning

Sarah Rathnam SARAH_RATHNAM@G.HARVARD.EDU
John A. Paulson School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA

Sonali Parbhoo S.PARBHOO@IMPERIAL.AC.UK
Imperial College London
London SW7 2BX, UK

Siddharth Swaroop SIDDHARTH@SEAS.HARVARD.EDU
Weiwei Pan WEIWEIPAN@G.HARVARD.EDU
Susan A. Murphy SAMURPHY@G.HARVARD.EDU
Finale Doshi-Velez FINALE@SEAS.HARVARD.EDU
John A. Paulson School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138 USA

Editor: Nan Jiang

Abstract

Discount regularization, using a shorter planning horizon when calculating the optimal policy, is a popular choice to avoid overfitting when faced with sparse or noisy data. It is commonly interpreted as de-emphasizing or ignoring delayed effects. In this paper, we prove two alternative views of discount regularization that expose unintended consequences and motivate novel regularization methods. In model-based RL, planning under a lower discount factor acts like a prior with stronger regularization on state-action pairs with more transition data. This leads to poor performance when the transition matrix is estimated from data sets with uneven amounts of data across state-action pairs. In model-free RL, discount regularization equates to planning using a weighted average Bellman update, where the agent plans as if the values of all state-action pairs are closer than implied by the data. Our equivalence theorems motivate simple methods that generalize discount regularization by setting parameters locally for individual state-action pairs rather than globally. We demonstrate the failures of discount regularization and how we remedy them using our state-action-specific methods across empirical examples with both tabular and continuous state spaces.

Keywords: reinforcement learning, regularization, certainty equivalence, discount factor, Markov decision process

1. Introduction

In reinforcement learning (RL), planning under a shorter horizon is a common form of regularization with a straightforward interpretation: restricting policy class complexity by optimizing for shorter term rewards (Jiang et al., 2015). In the most extreme case, planning with a discount factor of zero results in a contextual bandit algorithm. Using a reduced or zero discount factor for planning is common in real-world applications such as mobile health (Liao et al., 2020; Trella et al., 2022), medicine (Oh et al., 2022; Awasthi et al., 2022; Durand et al., 2018), and education (Cai et al., 2021; Qi et al., 2018), particularly in low-data settings.

In this paper, we demonstrate the equivalence between discount regularization and other common regularization techniques. These connections provide a deeper understanding of discount regularization that reveals its limitations. We first prove that in model-based tabular RL, discount regularization produces the same optimal policy as averaging the transition matrix with a regularization matrix that is the same for all states and actions. In this view, discount regularization forces the agent to plan as if the distribution over next state is more similar to one another across different states and actions than was observed in the data. This can also be viewed in terms of a prior on the transition matrix. We prove an analogous result for model-free RL: discount regularization produces the same optimal policy as a modified Q-function update rule of a specific weighted-average form. In this form, we see that discount regularization acts similarly to a penalty on the value function, forcing the state-action values to be more similar. Finally, we demonstrate mathematically that discount regularization approximates a λ-return.

Reframing discount regularization in these ways exposes unintended consequences. One consequence is that the magnitude of the prior implied by discount regularization is higher for state-action pairs with more transition observations in the data and vice versa. This is generally not desirable as we want stronger regularization on state-action pairs that we have observed less, and to rely on the data in those that we have observed more. Another negative effect is that the implied prior has the same transition distribution for all state-action pairs. This is inappropriate in many contexts, where it is better to replace this implicit prior with an informed prior that reflects knowledge about the environment.

Our equivalence theorems motivate model-free and model-based offline regularization methods that offer solutions to the problems exposed above. In the model-based setting, we apply *certainty-equivalence RL*, where the agent takes the estimated model as true when calculating the optimal policy (Goodwin and Sin, 1984). We take a simple estimate of the transition model (MLE for a tabular state space or kernel regression if continuous) and regularize the estimate to use within simple RL methods. Our methods tailor regularization to the task at hand, which includes both the data set and the environment. To mitigate the issue of inconsistent prior magnitudes in data sets with uneven exploration, we derive a state-action specific formula for the regularization parameter. The method we use to derive this parameter can be adapted to other priors to match the transition dynamics of the environment.

Finally, we demonstrate our results empirically. First, we confirm our equivalence theorems in a tabular setting. We then demonstrate that a uniform prior with fixed magnitude across state-action pairs outperforms discount regularization across environments. We compare our model-based and model-free state-action-specific regularization methods to each other as well as to regularization with a fixed global parameter. Then, we demonstrate that our model-based method extends successfully to a continuous state space.

This paper extends previous work published in Rathnam et al. (2023). The major extensions are (1) a model-free analysis that connects discount regularization to a penalized Q-function, (2) analysis of how discount regularization approximates a truncated lambda-return, (3) introduction, analysis and simulations of a novel model-free regularization method, including its relationship to pessimism, and (4) application of novel regularization methods to a continuous state space.

2. Related Works

Discount Regularization. Jiang et al. (2015) demonstrate that the optimal policy generated using a "planning discount factor" that is shorter than the true discount factor of the MDP often outperforms the policy learned using the true discount factor when both policies are evaluated in the true environment (using the true discount factor). They prove that using a lower planning discount factor to calculate the optimal policy controls model complexity by restricting the number of possible policies considered, thereby avoiding overfitting. They further demonstrate that the benefit of a lower planning discount factor is increasingly pronounced in cases where the model is estimated from a smaller data set. Amit et al. (2020) refer to this concept as "discount regularization," a term which we use here. Unlike these works, we provide means to connect discount regularization with placing a prior on the transition matrix.

Previous works also discuss the limitations of a fixed discount factor and present approaches for more flexible discounting, for example state-dependent (Wei and Guo, 2011; Yoshida et al., 2013), state-action-dependent (Pitis, 2019), and transition-based discounting (White, 2017). We add to this work by demonstrating that discount regularization carries implicit assumptions of equal transition distributions for all state-action pairs and stronger regularization on those with more transition data.

Bayesian RL. While a Bayesian prior encodes expert knowledge, information from previous studies, or other outside information, we can also view a prior as a form of regularization since it forces the model not to overfit when data is limited (Poggio and Girosi, 1990; Ghavamzadeh et al., 2015). This is a flexible tool that allows us to regularize in a way that matches our prior knowledge and beliefs about the environment. In model-based Bayesian RL, the problem is often framed as a Bayes-Adaptive MDP (BAMDP), an MDP where the states are replaced by "hyperstates" that reflect the original state space combined with the posterior parameters of the transition function (Duff, 2002). In general, Bayesian RL algorithms do not explicitly address planner overfitting; rather they incorporate the probability distribution over models, causing the planner not to overfit to an uncertain model. For example, model-based Bayesian RL methods draw sample models from the posterior (Asmuth et al., 2012), sample hyperstates (Poupart et al., 2006), or apply an exploration bonus based on the amount of data (Kolter and Ng, 2009a) or based on the variance of the parameters (Sorg et al., 2012). The BAMDP framework can also be extended to the case of partial observability (Ross et al., 2007, 2011). In this paper, we discuss planning using the posterior mean of the transition matrix under a Dirichlet prior as a regularized form of the transition matrix, which is a common choice in model-based RL, e.g. Vlassis et al. (2012); O'Donoghue et al. (2020). In contrast to Bayesian methods, the methods that we propose regularize the model to get a point estimate which can be used directly in simple non-Bayesian RL methods.

Penalized Value Function. In the model-free setting, we relate discount regularization to a modified value function. Previous RL regularization methods work by adding a bonus or penalty to the value function. For example, an L_1 (Kolter and Ng, 2009b; Liu et al., 2012; Ghavamzadeh et al., 2011) or L_2 (Farebrother et al., 2018; Cobbe et al., 2019) penalty is commonly added to the value function. Entropy regularization can function by adding an entropy bonus to the gradient of the Q-function (O'Donoghue et al., 2016) or the reward (Nachum et al., 2017). Another use of a value function bonus is to encourage exploration by adding a bonus to the value of unseen states (Kolter and Ng, 2009a). Conversely, offline methods often employ pessimism penalize the value of states and actions not well-explored in the data. We analyze our method in the context of point-wise pessimism Jin et al. (2021) and Bellman-consistent pessimism Xie et al. (2021) in Sec. 5.2.3.

Connecting Regularization Methods. Several previous works have established connections between regularization methods, as we do here. For example, Wu et al. (2019) introduce a framework that connects a penalty on value function with policy regularization, Neu et al. (2017) connect entropy-regularized algorithms, and Li et al. (2006) present a unified view of state aggregation schemes. Liu et al. (2019) empirically compare a wider range of regularizers. More recently, Eysenbach et al. (2023) prove that one-step RL and critic regularization methods result in the same policy under certain assumptions. Most similar to our work, Amit et al. (2020) connect discount regularization with L_2 regularization in TD learning. We discuss the connection to this work in Section 4.3.

3. Background and Notation

Markov Decision Process. We consider a finite, discrete Markov decision process (MDP). An MDP M is characterized by $< S, A, R, T, \gamma >$, defined as follows. S: State space of size N_s. A: Action space of size N_a. $R(s, a)$: Reward, as a function of state s and action a. $T(s, a, \cdot)$: Transition function, mapping each state-action pair to a probability distribution over successor states, assumed unknown and estimated from the data. γ: True discount factor for the MDP, $0 \leq \gamma < 1$, under which a policy π is evaluated. We also use the following notation: γ_p: Planning discount factor $0 \leq \gamma_p < \gamma$. γ_p is not used to evaluate the policy; it is used for planning only, where replacing γ with γ_p serves as a regularizer ; $c_{i,j,k}$: count of transition observations in data set starting at state s_i, taking action a_j and transitioning to state s_k.

Certainty Equivalence. The model-based portion of our analysis is in the context of certainty-equivalence RL. Certainty equivalence is a useful approach to model-based RL where the agent takes the estimated model as accurate when finding the optimal policy. It separates the estimation of the model from the policy optimization (Goodwin and Sin, 1984). The maximum likelihood estimate (MLE) is a natural choice for the model estimate, however maximum likelihood solutions can overfit, particularly in the case of small data sets (Murphy, 2012). Often, a better policy is obtained by regularizing the MDP before learning the certainty-equivalence policy.

4. Alternative Views of Discount Regularization and Connections to Other Regularization Methods

Discount regularization is a simple concept—the agent finds an optimal policy using a shorter horizon than the environment's true horizon γ—yet analyzing its relationship to other methods provides new insights on regularization. In the sections that follow, we will demonstrate how discount regularization relates to different methods and interpretations: a weighted average T, a Bayesian prior on T, a penalized Q-function, and λ-returns.

4.1 Discount Regularization as a Weighted Average Transition Matrix

We begin with a reframing of discount regularization as a weighted average transition matrix. This form illustrates the classic view of discounting as partial termination and motivates our first equivalence theorem.

4.1.1 Discount Regularization as Partial Termination

First, we show that discount regularization is mathematically equivalent to replacing the transition matrix with the weighted average between that transition matrix and a matrix of zeros. The form is unusual as the matrix of zeros is not a transition matrix, however it gives intuition on discount regularization and motivates the equivalence theorem and Bayesian formulation that follow.

To cast discount regularization in certainty-equivalence RL as a weighted average transition matrix, consider the Bellman equation for the value of each state under policy π, $V^\pi = R_\pi + \gamma T_\pi V^\pi$, where the vector V^π is the value of each state, R_π is the vector of rewards, and T_π is the transition matrix, all under policy π. Let $\gamma_p < \gamma$ be the planning discount factor, the lower discount factor used for regularization when calculating the certainty-equivalence policy. (This policy will be evaluated under the true discount factor γ.) Then we have the Bellman equation $V^\pi = R_\pi + \gamma_p T_\pi V^\pi$. We rewrite the product $\gamma_p T_\pi$ from the Bellman equation as the product of true discount factor γ and a weighted average matrix: $\gamma_p T_\pi = \gamma[(1-\epsilon)T_\pi + \epsilon T_{\text{zeros}}]$, where T_{zeros} is an appropriately sized matrix of zeros and $\epsilon = \frac{\gamma-\gamma_p}{\gamma}$.

Using this insight, when estimating the transition matrix from data, we can use the following weighted average transition matrix and the true discount factor γ for planning in place of the MLE transition matrix and lower discount factor γ_p.[1]

$$\hat{T}_{\substack{\text{disc}\\\text{reg}}}(s_i, a_j, \cdot) = (1-\epsilon)\hat{T}_{\text{MLE}}(s_i, a_j, \cdot) + \epsilon T_{\text{zeros}}, \qquad \text{where } \epsilon = \frac{\gamma - \gamma_p}{\gamma}. \tag{1}$$

In Theorem 1, we will broaden the relationship between discount regularization and a weighted average transition matrix demonstrated here. We prove that discount regularization generates the same optimal policy as replacing the transition matrix with a weighted average matrix of a specific form.

Eq. 1 provides another way to view discounting as "partial termination" (Sutton and Barto, 2018). According to this classic interpretation, the sum of discounted rewards can be viewed as the sum of undiscounted rewards partially terminating with degree 1 minus

[1] See Appendix A.2 for details.

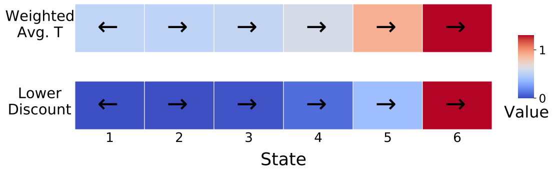


Figure 1: River Swim MDP described in Sec 6.1. Planning with lower discount rate or weighted average T yield different values (colors), but the same optimal policy (arrows).

the discount factor at each step. This can also be viewed as the agent transitioning to an absorbing state with probability 1 minus the discount factor at each step. To see this, observe that setting $\gamma = 1$, Eq. 1 represents the agent terminating with probability $1 - \gamma_p$ at each step.

4.1.2 Equivalent Policies Regularizing the Horizon and Transition Matrix

The relationship between discount regularization and a weighted average transition matrix is in fact more general than discussed in the previous section. Eq. 1 shows that discount regularization is mathematically equivalent to averaging the transition matrix with a matrix of zeros, but in fact it also produces the same optimal policy as averaging the transition matrix with any regularization matrix that is the same for all states and actions when both methods use the same value of ϵ. This result is stated more precisely in Thm. 1 and illustrated in Fig. 1.

Theorem 1 *Let M_1 and M_2 be finite-state, infinite horizon MDPs with identical state space, action space, reward function. Let $0 < \gamma < 1$, $0 < \epsilon \leq 1$, and let $T_{reg}(s, a, \cdot)$ be any matrix used for regularization that is the same for all (s,a), i.e. $T_{reg}(s, a, \cdot) = \vec{v} \quad \forall (s, a)$.*

If M_1 has transition function T and uses discount rate $(1 - \epsilon)\gamma$ in planning and M_2 has transition function $(1 - \epsilon)T + \epsilon T_{reg}$, and uses discount rate γ in planning, then M_1 and M_2 have the same optimal policy.

Proof

(1) *The optimal policy for all MDPs whose Bellman optimality equations differ only by added constant c to the reward are the same.* Consider Bellman's optimality equation for any arbitrary state s and action a for an MDP in which constant c is added to every reward $R(s, a)$:

$$Q^*(s, a) = R(s, a) + c + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q^*(s', a').$$

It is known that the optimal policy of an MDP is not affected by adding the same constant c to all rewards $R(s, a)$. (See, for example, Ng et al. (1999): "constant offsets of the reward do not affect the optimal policy when $\gamma < 1$".) Proof of this step in Appendix B.

It follows that the optimal policy $\pi_{\text{opt}}(s) = \text{argmax}_a Q^*(s,a)$ is the same for all values of c. So for all values of constant c, the MDP with the Bellman optimality equation above has the same optimal policy.

(2) *The Bellman optimality equation for an MDP in which the transition matrix is regularized by taking its weighted average with a matrix T_{reg} can be written in terms of a lower discount factor and an added constant.* Let $T_{reg}(s,a,\cdot)$ be a transition matrix that is the same for all (s,a),

$$\begin{aligned} Q^*(s,a) &= R(s,a) + \gamma \sum_{s'} [((1-\epsilon)T(s,a,s') + \epsilon T_{reg}(s,a,s')) \max_{a'} Q^*(s',a')] \\ &= R(s,a) + \gamma(1-\epsilon) \sum_{s'} T(s,a,s') \max_{a'} Q^*(s',a') + \gamma\epsilon \sum_{s'} T_{reg}(s,a,s') \max_{a'} Q^*(s',a'). \end{aligned}$$

Letting $c(s,a) = \gamma\epsilon \sum_{s'} T_{reg}(s,a,s') \max_{a'} Q^*(s',a')$, Bellman's optimality equation is:

$$Q^*(s,a) = R(s,a) + c(s,a) + \gamma(1-\epsilon) \sum_{s'} T(s,a,s') \max_{a'} Q^*(s',a').$$

By the assumptions of Thm. 1, $T_{reg}(s,a,s')$ is the same for all (s,a) and is therefore a function of s' only. $\max_{a'} Q^*(s',a')$ is also a function of s' only. Therefore $c(s,a)$ is a number, which we can call c,

$$c = \gamma\epsilon \sum_{s'} \underbrace{T_{reg}(s,a,s')}_{\text{func. of s' only}} \underbrace{\max_{a'} Q^*(s',a')}_{\text{func. of s' only}} = \text{constant}.$$

(3) *Setting constant c to 0 does not change the optimal policy of the resulting MDP.* By (1), replacing c with 0, the resulting new MDP with Bellman optimality equation

$$Q^*(s,a) = R(s,a) + \gamma(1-\epsilon) \sum_{s'} T(s,a,s') \max_{a'} Q^*(s',a')$$

has the same optimal policy as the MDP whose Bellman optimality equation has constant c added to the reward.

(4) *The resulting Bellman equation is that of an MDP with the original unregularized transition matrix $T(s,a,s')$ and reduced discount factor $(1-\epsilon)\gamma$.* Therefore, the MDP with discount rate γ and transition matrix $(1-\epsilon)T(s,a,s') + \epsilon T_{reg}(s,a,s')$ and the MDP with discount rate $\gamma(1-\epsilon)$ and transition matrix $T(s,a,s')$ have identical optimal policies. ■

Thm. 1 provides a deeper understanding of how discount regularization functions. At maximum regularization, $\gamma_p = 0$ or equivalently $\epsilon = 1$, it unites two views of the relationship between bandit and MDP algorithms. An MDP algorithm with $\gamma = 0$ creates a (non-adversarial) contextual bandit algorithm (Agarwal et al., 2019). Alternatively, when "the transition probability is identical... for all states and actions" in an MDP algorithm, it also forms a contextual bandit algorithm (Zanette and Brunskill, 2018). Our proof extends this equivalence beyond the bandit setting to all amounts of regularization.

Thm. 1 also reveals the limitations of discount regularization. First, the regularization matrix is the same for all state-action pairs, so it will be biased in environments where the distribution over next state varies greatly across state-action pairs. Furthermore, as we demonstrate in Sec. 4.2, this theorem leads to the result that discount regularization provides stronger regularization on state-action pairs with more data.

4.2 Discount Regularization Implies a Dirichlet Prior on the Transition Function

As discussed in Sec. 2, a Dirichlet prior on the transition matrix T functions as a flexible form of regularization. Given a prior on T for state-action pair (s_i, a_j), $T_{\text{prior}}(s_i, a_j, \cdot) \sim \text{Dirichlet}(\alpha_{i,j,1}, ..., \alpha_{i,j,N_s})$, the posterior mean represents a regularized form. Though simple, this generates several important insights that deepen our understanding and facilitate better regularization.

4.2.1 Posterior Mean as a Weighted Average

Let $\langle c_{i,j,1}, ..., c_{i,j,N_s} \rangle$ be the transition count data observed from state s_i to states 1 through N_s under action a_j. Then the posterior mean of the transition matrix, $\hat{T}_{\substack{\text{post}\\\text{mean}}}$, is equal to a weighted average of the MLE transition matrix and the mean of the prior:[2]

$$\begin{gathered}\hat{T}_{\substack{\text{post}\\\text{mean}}}(s_i, a_j, \cdot) = (1 - \epsilon_{i,j})\hat{T}_{MLE}(s_i, a_j, \cdot) + \epsilon_{i,j} T_{\substack{\text{prior}\\\text{mean}}}(s_i, a_j, \cdot), \\ \epsilon_{i,j} = \frac{\sum_{k=1}^{N_s} \alpha_{i,j,k}}{\sum_{k=1}^{N_s} c_{i,j,k} + \sum_{k=1}^{N_s} \alpha_{i,j,k}}.\end{gathered} \tag{2}$$

4.2.2 Deriving the Prior Magnitude

In Thm. 1, we proved that discount regularization produces the same optimal policy as averaging the transition matrix with any regularization matrix that is the same for all states and actions. We also know from Eq. 2 that a weighted average transition matrix can be written in terms of the MLE transition matrix and a Dirichlet prior. In this section, we combine these two relationships to show that using state-action visitation rates from the data allows us to produce an empirical Bayes prior on $T(s, a, \cdot)$ that results in the same optimal policy as discount regularization.

Adapting the form of Thm. 1 to the setting where T is estimated as the MLE of the transition data, discount regularization with planning discount factor $\gamma_p < \gamma$ produces the same optimal policy as replacing $\hat{T}_{MLE}(s, a, \cdot)$ with $(1 - \epsilon)\hat{T}_{MLE} + \epsilon T_{reg}$, where $\epsilon = \frac{\gamma - \gamma_p}{\gamma}$ and T_{reg} is the same for all (s, a). Using Eq. 2, we can view this weighted average transition matrix as a posterior mean. Viewing $(1 - \epsilon)\hat{T}_{MLE} + \epsilon T_{reg}$ as a Bayesian posterior mean, the weight $\epsilon = \frac{\gamma - \gamma_p}{\gamma}$ equates to $\epsilon_{i,j}$ in Eq. 2.

Since discount regularization employs the same planning discount rate and consequently the same value of ϵ for every state-action pair, the prior that produces an equivalent policy also has the same value of ϵ at every state-action pair. Setting the formulas for ϵ from Eq. 1 and Eq. 2 equal and solving for the sum of the prior magnitude $\sum_{k=1}^{N_s} \alpha_{i,j,k}$ reveals the relationship between the planning discount factor γ_p and the prior magnitude. We see that a lower planning discount factor implies a prior whose magnitude depends on the number

[2] See Appendix A.1 for derivation.

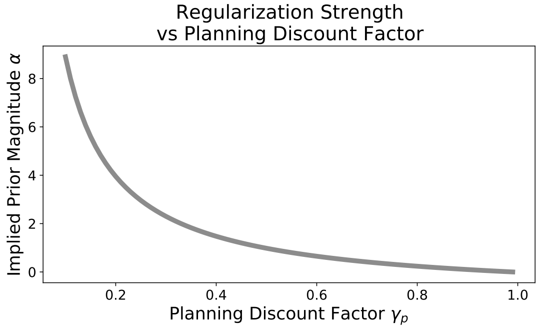


Figure 2: Smaller planning discount factor γ_p implies larger magnitude of a uniform Dirichlet prior for an MDP with 10 states, 20 transition observations per state, and γ = 0.99.

of transitions from (s_i, a_j) in the data, [3]

$$\sum_{k=1}^{N_s} \alpha_{i,j,k} = \left(\frac{\gamma - \gamma_p}{\gamma_p}\right) \sum_{k=1}^{N_s} c_{i,j,k}. \tag{3}$$

In the case of a uniform prior, which we use in our simulations, the magnitude simplifies to

$$\alpha_{i,j,k} = \left(\frac{\gamma - \gamma_p}{\gamma_p}\right) \frac{\sum_{k=1}^{N_s} c_{i,j,k}}{N_s} \quad \forall k.$$

The relationship between uniform prior magnitude $\alpha_{i,j,k}$ and planning discount factor γ_p for an individual state-action pair is illustrated in Fig. 2 where a smaller planning discount factor γ_p implies a larger uniform prior magnitude α. Furthermore, Eq. 3 shows us that, for any planning discount factor γ_p, the magnitude of the corresponding Dirichlet prior is higher for state-action pairs with more data. In other words, those (s, a) with more observations in the data are regularized more. Especially for data sets with uneven distribution of transition data, it may be better to use a more flexible regularization method. In Sec. 5.1, we introduce state-action-specific regularization to mitigate this issue. Note that the special case of $\gamma_p = 0$, a type of contextual bandit, presents an exception as the implied priors for all (s, a) are of infinite magnitude. This case is fundamentally different as the future is not just discounted but rather completely ignored.

Next we demonstrate that the same equivalence relationship from Thm. 1 holds in the model-free setting. We will see that by changing the setting from model-based to model-free, discount regularization can be viewed as a modified Q-function update rule instead of a prior on the transition matrix.

4.3 Discount Regularization as a Modified Q-Function

The equivalence between regularizing the horizon and regularizing the transition matrix arises when viewed in the model-based setting, however, this relationship extends to model-

[3] See Appendix A.3 for details.

free algorithms as well. Restated in the model-free context, discount regularization functions like an added penalty or bonus on the Q-function.

Theorem 2 *Let M_1 and M_2 be finite-state, infinite horizon MDPs with identical state space, action space, and reward function. Let discount factor $0 \leq \gamma < 1$, regularization parameter $0 < \epsilon \leq 1$, and let $Q_{reg}(s,a)$ be a function used to regularize the Q-function that is constant in (s,a), i.e. $Q_{reg}(s,a) = Q_{reg}$.*

If M_1 uses discount rate $\gamma_1 = (1-\epsilon)\gamma$ and state-action value function

$$Q^*(s,a) = R(s,a) + \gamma_1 E_{s' \sim T(S,A,\cdot)}[\max_{a'} Q^*(s',a')|S=s, A=a]$$

in planning, and M_2 uses discount rate γ and state-action value function

$$Q^*(s,a) = R(s,a) + \gamma E_{s' \sim T(S,A,\cdot)}\big[\max_{a'}[(1-\epsilon)Q^*(s',a') + \epsilon Q_{reg}(s,a)]|S=s, A=a\big] \quad (4)$$

in planning, then M_1 and M_2 have the same optimal policy.

Proof **(1)** $Q_{reg}(s,a)$ does not depend on T, so Eq. 4 is equal to:

$$Q^*(s,a) = R(s,a) + \gamma\epsilon Q_{reg}(s,a) + \gamma E_{s' \sim T(S,A,\cdot)}\big[\max_{a'}(1-\epsilon)Q^*(s',a')|S=s, A=a\big].$$

(2) $Q_{reg}(s,a)$ is constant by construction, therefore $c = \gamma\epsilon Q_{reg}(s,a) = \gamma\epsilon Q_{reg}$ is constant,

$$Q^*(s,a) = R(s,a) + c + \gamma E_{s' \sim T(S,A,\cdot)}\big[\max_{a'}(1-\epsilon)Q^*(s',a')|S=s, A=a\big].$$

(3) By step (1) of the proof of Thm. 1, setting $c = 0$ does not change the optimal policy. Therefore the MDP with the following Q-function has the same optimal policy as M_2,

$$\begin{aligned} Q^*(s,a) &= R(s,a) + \gamma E_{s' \sim T(S,A,\cdot)}\big[\max_{a'}(1-\epsilon)Q^*(s',a')|S=s, A=a\big] \\ &= R(s,a) + \gamma(1-\epsilon) E_{s' \sim T(S,A,\cdot)}\big[\max_{a'} Q^*(s',a')|S=s, A=a\big] \\ &= R(s,a) + \gamma_1 E_{s' \sim T(S,A,\cdot)}\big[\max_{a'} Q^*(s',a')|S=s, A=a\big]. \end{aligned}$$

This is the Q-function for MDP M_1. ■

The model-based (Thm. 1) and model-free (Thm. 2) versions of the discount regularization equivalence theorem make the same assumption: the agent plans as if it transitions according to a distribution $T_{reg}(s,a,s')$ that is the same for all (s,a) with probability ϵ at each step. This is clear in the model-based version. In the model-free setting, transitioning according to T_{reg} at each step with probability ϵ equates to averaging the expected value of next state with $Q_{reg} = E_{s' \sim T_{reg}(S,A,\cdot)}[\max_{a'} Q^*(s',a')]$. Despite the same assumptions, the model-based and model-free versions lead to distinct interpretations. We showed above that when viewed in the model-based setting, discount regularization functions by restricting model complexity, or acting like a prior on the transition matrix. In the model-free setting, discount regularization relates to regularization methods that penalize the value function.

As mentioned in Sec. 2, previous works have proven equivalences between different regularization methods that place a penalty or bonus on the value function. The equivalence in Thm. 2 is comparable to Proposition 1 in Amit et al. (2020). They showed that discount regularization in TD learning methods functions equivalently to a regularization term added to the objective. The formula for the regularization term differs from ours given the different setting and assumptions, however the insights are consistent. First, they note that, "This term penalizes large value estimates and therefore encourages consistent value estimates across state-action pairs which may encourage generalization by reducing the effect of spurious approximation errors." In our Thm. 2, the modified Q-function averages the state-action value with a constant, pushing all value estimates closer together. They also conclude that "...states that are visited less often are less regularized" because the regularization term depends on the distribution of states in the data. This is exactly what we observed from the empirical Bayes prior in the model-based setting as discussed in Sec. 4.2.

4.4 Discount Regularization as a Truncated Lambda Return

A final interpretation of discount regularization is as an approximation to the lambda return, which is another common form of regularization in RL.

Discount regularization calculates an approximate value equal to a truncated lambda return. Unlike the λ-return, discount regularization does not give the exact return for a fixed $\gamma_p < \gamma$. By expanding out the terms of the λ-return, we show that the approximate return under discount regularization is equal to a λ-return with truncated k-step returns.

To see this, first, expand both sums in the definition of the λ-return.

$$\begin{aligned}
R_t^\lambda &= (1-\lambda)\sum_{k=1}^{\infty}\lambda^{k-1}R_t^{(k)} \\
&= (1-\lambda)\sum_{k=1}^{\infty}\lambda^{k-1}[\sum_{j=0}^{k-1}\gamma^j R_{t+j+1} + \gamma^k V^\pi(S_{t+k})] \\
&= (1-\lambda)\lambda^0[\gamma^0 R_{t+1} + \gamma^1 V^\pi(S_{t+1})] \\
&+ (1-\lambda)\lambda^1[\gamma^0 R_{t+1} + \gamma^1 R_{t+2} + \gamma^2 V^\pi(S_{t+2})] \\
&+ (1-\lambda)\lambda^2[\gamma^0 R_{t+1} + \gamma^1 R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V^\pi(S_{t+3})] \\
&+ \cdots
\end{aligned}$$

Truncating each k-step return after the k rewards (or equivalently setting $V^\pi(S_{t+k}) = 0 \forall k$), we recover discount regularization. To see this, group the terms for each reward together then set $V^\pi(S_{t+k}) = 0$.

First grouping terms:

$$
\begin{aligned}
R_t^\lambda &= (1-\lambda)\gamma^0(\sum_{k=0}^{\infty}\lambda^k)R_{t+1} + (1-\lambda)\gamma^1(\sum_{k=1}^{\infty}\lambda^k)R_{t+2} + (1-\lambda)\gamma^2(\sum_{k=2}^{\infty}\lambda^k)R_{t+3} + \cdots \\
&= \gamma^0(\lambda^0 + \lambda^1 + \lambda^2 + \cdots - \lambda^1 - \lambda^2 - \lambda^3 - \cdots)R_{t+1} \\
&+ \gamma^1(\lambda^1 + \lambda^2 + \lambda^3 + \cdots - \lambda^2 - \lambda^3 - \lambda^4 - \cdots)R_{t+2} \\
&+ \gamma^2(\lambda^2 + \lambda^3 + \lambda^4 + \cdots - \lambda^3 - \lambda^4 - \lambda^5 - \cdots)R_{t+3} \\
&+ \cdots \\
&+ (1-\lambda)\lambda^0\gamma^1 V^\pi(S_{t+1}) + (1-\lambda)\lambda^1\gamma^2 V^\pi(S_{t+2}) + (1-\lambda)\lambda^2\gamma^3 V^\pi(S_{t+3}) + \cdots \\
&= (\gamma\lambda)^0 R_{t+1} + (\gamma\lambda)^1 R_{t+2} + (\gamma\lambda)^2 R_{t+3} + \cdots + (1-\lambda)\gamma\sum_{k=0}^{\infty}(\lambda\gamma)^k V^\pi(S_{t+k+1})
\end{aligned}
$$

$$
R_t^\lambda = \sum_{k=0}^{\infty}(\gamma\lambda)^k R_{t+k+1} + (1-\lambda)\gamma\sum_{k=0}^{\infty}(\lambda\gamma)^k V^\pi(S_{t+k+1}) \tag{5}
$$

Set $V^\pi(S_{t+k}) = 0$ to get a truncated λ-return: $\hat{R}_t^\lambda = \sum_{k=0}^{\infty}(\gamma\lambda)^k R_{t+k+1}$. Let planning discount factor $\gamma_p = \gamma\lambda$. Then our truncated lambda return is equal to the return under discount regularization, $\hat{R}_t^\lambda = \sum_{k=0}^{\infty}\gamma_p^k R_{t+k+1}$.

Bias of the discount regularized value Since R_t^λ is equal to the true return, Eq. 5, decomposes the return into the discount regularized return and a bias term. Taking the expectation over policy π to get value, the bias of the discount regularized estimate of value is $\text{Bias}(V_{\substack{\text{disc}\\\text{reg}}}(s)) = -\mathbb{E}_\pi[(1-\lambda)\gamma\sum_{k=0}^{\infty}(\lambda\gamma)^k V^\pi(S_{t+k+1})|S_t = s]$. The bias is the weighted average value of states expected to be visited when following policy π starting at state s, with weight decaying over time. As expected, the discount regularized value is most biased for states from which it is expected to reach high-value states soon.

The truncated lambda return also provides another illustration of the interpretation of discounting as partial termination or transitioning to an absorbing state with probability 1 minus the discount factor at each step, as we saw in Sec. 4.1.1. Setting all state values in the bias term to 0 represents a partial termination at each step. In the undiscounted setting, $\gamma = 1$ and the bias term reflects partially terminating with degree $1-\lambda$ at each time step.

5. State-Action-Specific Regularization

We exposed in Eq. 3 that discount regularization functions like an empirical Bayes prior with the undesirable property of stronger regularization strength on state-action pairs with more data. To address this problem, we apply the model-free and model-based equivalence theorems to motivate methods that tailor the amount of regularization to each state-action pair rather than setting a global regularization parameter.

5.1 Model-Based Regularization Method

To avoid the issue of mismatched regularization strengths across state-action pairs, we return to the weighted average form introduced in Sec. 4.1.1 to derive a formula for state-action-specific regularization. Using this form, we calculate the MSE of the estimated transition matrix and identify the value of regularization parameter ϵ that minimizes this error separately for each (s, a). While we recognize that a low-MSE transition matrix estimate does not guarantee a good policy (since some errors in T affect the optimal policy more than others),[4] it is a reasonable step towards that goal.

We derive a closed-form expression for the MSE for the case of a uniform Dirichlet prior. We take $\text{MSE}(\hat{T}(s, a, \cdot))$ to be the sum of the MSE of the individual elements. The derivation using the bias-variance decomposition of MSE is provided in Appendix C and the resulting form is below. Letting $\hat{T}_{\text{unif}}$ be the posterior mean of T under a uniform Dirichlet prior,

$$\begin{aligned}\text{MSE}[\hat{T}_{\text{unif}}(s_i, a_j, \cdot)] = \sum_{k=1}^{N_s} &\underbrace{(1-\epsilon_{i,j})^2 \frac{1}{\sum_{k=1}^{N_s} c_{i,j,k}} T(s_i, a_j, s_k)(1 - T(s_i, a_j, s_k))}_{variance} \\ &+ \underbrace{\epsilon_{i,j}^2 \left(\frac{1}{N_s} - T(s_i, a_j, s_k)\right)^2}_{bias^2}.\end{aligned} \tag{6}$$

Let $\epsilon^*_{i,j}$ be the value of the regularization parameter $\epsilon_{i,j}$ that minimizes the MSE equation. Then,

$$\epsilon^*_{i,j} = \frac{K(s_i, a_j)}{K(s_i, a_j) + \sum_{k=1}^{N_s} c_{i,j,k}}, \quad \text{where } K(s_i, a_j) = \frac{\sum\limits_{k=1}^{N_s} T(s_i, a_j, s_k)(1 - T(s_i, a_j, s_k))}{\sum\limits_{k=1}^{N_s} (\frac{1}{N_s} - T(s_i, a_j, s_k))^2}. \tag{7}$$

The first term of Eq. 6 is the contribution of the MLE's variance to the error, in this case the only source of variance. The second term is the bias introduced by regularization. The strength of regularization $\epsilon_{i,j}$ controls the trade-off between the bias and variance. The variance is driven by the amount of data $\sum_{k=1}^{N_s} c_{i,j,k}$ both through its role in setting the amount of regularization and as a factor inversely impacting the variance term. Both bias and variance are impacted by the true transition distribution $T(s_i, a_j, \cdot)$. A deterministic $T(s_i, a_j, \cdot)$ maximizes bias for a given $\epsilon_{i,j}$, but results in $\epsilon^*_{i,j} = 0$ (since $T(s_i, a_j, s'_k)(1 - T(s_i, a_j, s'_k)) = 0$ for all s'_k). At the other extreme, if $T(s_i, a_j, \cdot)$ is uniform, variance is maximized for a given $\epsilon_{i,j}$ but there is no bias, so we default to $\epsilon^*_{i,j} = 1$. Intermediate values of $\epsilon^*_{i,j}$ trade off between bias and variance. A uniform prior on $T(s_i, a_j, \cdot)$ with state-action-specific parameter $\epsilon^*_{i,j}$ improves upon discount regularization by setting the parameters locally for each state-action pair rather than forcing one global regularization parameter.

Furthermore, there is no parameter tuning required, simply a plugin estimate for T (e.g. the MLE). In practice, the true transition matrix T is not known and must be estimated. We

[4] Consider, for example, the case of the River Swim Environment described in Sec. 6.1. Under the optimal policy, the agent goes right towards the large reward. Overestimating the probability of going right under the solid line action results in the agent preferring the correct action *more* and hence results in learning the correct optimal policy. Conversely, underestimating the probability of going right by a smaller amount could result in learning the wrong optimal policy despite lower MSE.

may worry that in the low data regimes in which regularization is required, the estimate of T will not be good enough to estimate $\epsilon^*_{i,j}$. Nonetheless, our empirical examples in Sec. 6.3 demonstrate that our formula for $\epsilon^*_{i,j}$ leads to a reduction in loss over a single global regularization parameter.

Note that the state-action-specific parameter $\epsilon^*_{i,j}$ combined with regularization matrix T_{reg} does not map directly to a state-action-specific discount factor. Step (2) of the proof of Thm. 1 depends on $c = \gamma\epsilon \sum_{s'} T_{reg}(s, a, s') \max_{a'} Q^*(s', a')$ being constant. Otherwise, we cannot set $c = 0$ and expect the resulting MDP (which represents discount regularization) to have the same optimal policy. A state-action-specific discount factor breaks this equivalence.

5.2 Model-Free Regularization Methods

In the model-free setting, as in the model-based setting, a fixed universal regularization parameter like the on used in discount regularization can cause a mismatch in regularization strengths when the count data is uneven across state-action pairs. To address this problem, we set regularization strength separately for each (s_i, a_j) based on the Q-function and amount of data. A natural choice for state-action specific parameter $\epsilon^*_{i,j}$ in the model-free setting is the value that minimizes the mean squared error in Q separately for each state-action pair. We investigate this approach, calculating the analytical solution by two different methods.

We demonstrate the performance of these two calculations for $\epsilon^*_{i,j}$ on fitted Q-iteration (FQI, Ernst et al. 2005). In FQI, $Q(s, a)$ is estimated for each observation $\{s, a, s'\}_d$ in the data set, $q_d = R(s_d, a_d) + \gamma \max_a \hat{Q}(s'_d, a)$.[5] FQI alternates between using $\hat{Q}$ to calculate $\{q\}_d$ and updating $\hat{Q}$ to minimize the MSE across $\{q\}_d$. To regularize, we use the weighted average Q-function from Eq. 4 in calculating $\{q\}_d$, with Q_{reg} equal to the average value across states. This is equivalent to the case of a uniform regularization matrix $T_{reg}(s_i, a_j, \cdot)$ in the model-based setting. We update $\epsilon^*_{i,j}$ at each iteration, minimizing the sum of errors in estimates of $Q(s_i, a_j)$ across the data. The procedure is detailed in Algorithm 1.

5.2.1 Regularization Parameter Calculation: Method 1

To derive $\epsilon^*_{i,j}$, we first calculate the sum of squared errors (SSE). Given the iterative procedure, we derive the SSE and resulting expression for $\epsilon^*_{i,j}$ for a fixed policy π and associated Q-function $Q^\pi(s_i, a_j)$, which is updated at every step.

The SSE is the sum across all data tuples of the squared errors incurred by estimating the Q-function for tuple $\{s_i, a_j, s'_k\}$ with weighted-average Q-function $R(s_i, a_j) + \gamma(1 - \epsilon_{i,j})Q^\pi(s_i, \pi(s_i)) + \gamma\epsilon_{i,j}\frac{1}{N_s}\sum_{k'=1}^{N_s} Q^\pi(s_{k'}, \pi(s_{k'}))$. Therefore, the contribution of each state-action pair to the error calculation is weighted by how frequently it appears in the data. This results in the expression,

$$\begin{aligned} SSE = \sum_{i,j}\sum_{k=1}^{N_s} c_{i,j,k}\big(R(s_i, a_j) + \gamma(1 - \epsilon_{i,j})Q^\pi(s_k, \pi(s_k)) \\ + \gamma\epsilon_{i,j}\frac{1}{N_s}\sum_{k'=1}^{N_s} Q^\pi(s_{k'}, \pi(s_{k'})) - Q^\pi(s_i, a_j)\big)^2. \end{aligned} \tag{8}$$

[5] For simplicity, assume rewards are known.

Algorithm 1 FQI with State-Action-Specific Regularization

```
Initialize $\hat{Q}(s,a) = 0 \forall (s,a)$
Initialize $\epsilon^*(s,a)$ randomly
for i in num_iters do
    Start with P = {}
    Step 1: Estimate Q for each (s,a) data observation
    for every tuple {s,a,s'} in data set do
        $q = R(s,a) + \gamma(1-\epsilon^*_{s,a}) \max_a \hat{Q}(s',a) + \gamma\epsilon^*_{s,a} \text{average}_{s''}(\max_{a'} \hat{Q}(s'',a'))$
        Add {(s,a),q} to P
    end for
    Step 2: Learn Q
    Fit linear regression model with one-hot (s,a) encoding as features and q as dependent variable
    Set $\hat{Q}(s,a)$ equal to the coefficient for (s,a) from the regression model
    Step 3: Update $\epsilon^*_{s,a}$. Set $\epsilon^*_{s,a}$ equal to the diagonal values of matrix $\mathcal{E}$ calculated by Eq 9, using $\hat{Q}$ from Step 2 and $\pi^*(s) = argmax_a \hat{Q}(s,a)$.
end for
return value, optimal policy based on learned Q
```

For each (s_i, a_j), we solve for the value $\epsilon^*_{i,j}$ that minimizes Eq. 8 by setting the derivative equal to zero. Then we re-write the expression in matrix form to solve for all $\epsilon^*_{i,j}$ simultaneously. We use the following notation for the matrix equation: $\mathcal{E}$ is an $N_sN_a \times N_sN_a$ matrix with $\epsilon_{i,j}$ across the main diagonal, v^π_{avg} is the average state value under policy π, C is the $N_sN_a \times N_s$ matrix of transition counts (one row for every (s,a), one column for every s'), Π is a matrix mapping Q^π to V^π for fixed policy π. We get,

$$\begin{aligned}\mathcal{E} = \Big[&\gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}}) - 2\gamma^2 v^\pi_{avg} Diag(C\Pi^T Q^\pi) + \gamma^2 (v^\pi_{avg})^2 Diag(C\mathbf{1}_{N_s\times 1})\Big]^{-1} \\ &\Big[\gamma Diag(R) C\Pi^T Q^\pi + \gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}})\mathbf{1}_{N_sN_a\times 1} - \gamma Diag(C\Pi^T Q^\pi) Q^\pi \\ &- \gamma v^\pi_{avg} Diag(R) C\mathbf{1}_{N_s\times 1} - \gamma^2 v^\pi_{avg} Diag(C\Pi^T Q^\pi)\mathbf{1}_{N_sN_a\times 1} + \gamma v^\pi_{avg} Diag(C\mathbf{1}_{N_s\times 1}) Q^\pi\Big].\end{aligned} \tag{9}$$

Derivations for both SSE and $\mathcal{E}$ are in Appendix D.

5.2.2 Regularization Parameter Calculation: Method 2

We also calculate the MSE of $Q^\pi(s,a)$ using the assumption that the transition count data follows a multinomial distribution, similar to the model-based case. For each (s,a), we separately calculate the bias and variance resulting from estimating the Q-function for tuple $\{s_i, a_j, s'_k\}$ with weighted-average Q-function

$$R(s_i, a_j) + \gamma(1 - \epsilon_{i,j}) \sum_{k=1}^{N_s} T(s_i, a_j, s_k) Q^\pi(s_k, \pi(s_k)) + \gamma\epsilon_{i,j} \frac{1}{N_s} \sum_{k'=1}^{N_s} Q^\pi(s_{k'}, \pi(s_{k'})).$$

Like in the model-based case, T is estimated from the transition counts in the data, which is assumed to be multinomially distributed. In this case, each state-action pair contributes equally to the error, rather than each data tuple contributing equally as method 1. The MSE derivation for this method is in Appendix D.3.

These methods perform similarly to each other in our simulations, but do not perform as well as the model-based approach or even a fixed global regularization parameter. We discuss the reasons in Sec. 6.4.3.

5.2.3 Connections to Pessimism

By setting regularization strength inversely proportional to the transition count in the data, our method resembles pessimism however, it does not satisfy the definitions of point-wise or Bellman-consistent pessimism in Jin et al. (2021) and Xie et al. (2021), respectively.

Point-wise Pessimism. In the pessimistic value iteration meta-algorithm introduced in Jin et al. (2021), pessimism is encoded as a state-action-specific penalty applied to the Bellman operator at each step of value iteration. Specifically, this penalty is instantiated by subtracting a ξ-uncertainty quantifier $\Gamma(s, a)$, a function that bounds the absolute value of the difference between the true Bellman operator and its estimate using the data with probability $1 - \xi$.[6]

The Bellman operator $\hat{\mathbb{B}}_{\epsilon^*}\hat{V}(s, a)$ in our method can be framed similarly. In the model-free setting, we have:

$$\begin{aligned}
\hat{\mathbb{B}}_{\epsilon^*}\hat{V}(s, a) &= R(s, a) + \gamma \max_{a'}((1 - \epsilon^*_{s,a})\hat{Q}(s', a') + \epsilon^*_{s,a} Q_{reg}) \\
&= \underbrace{R(s, a) + \gamma \max_{a'} \hat{Q}(s', a')}_{\hat{\mathbb{B}}\hat{V}(s,a)} + \underbrace{\gamma\epsilon^*_{s,a}(Q_{reg} - \max_{a'} \hat{Q}(s', a'))}_{\text{penalty/bonus}}
\end{aligned}$$

We must have $\Gamma(s, a) \geq 0$ in order to be an uncertainty quantifier because it upper bounds an absolute value. This would equate to a negative or zero "penalty/bonus" term in the equation above. Our penalty term, however, can be either positive or negative, pushing estimated Q-values towards Q_{reg} rather than strictly decreasing them based on uncertainty. Thus it does not fit the definition and our method is not pointwise pessimistic. Furthermore, because the penalty term in our method is not a ξ-uncertainty quantifier, we cannot apply the sub-optimality guarantee in Theorem 4.2 of Jin et al. (2021).

The performance of pessimistic value iteration depends on finding an uncertainty quantifier that tightly bounds the error in the Bellman operator, but in practice this may be difficult to find. While our method does not come with performance guarantees, it provides a methodology for how to compute a penalty or bonus for the value equation. We do note,

[6] Please see Def. 4.1 of Jin et al. (2021) for formal definition.

however, that a weakness of our method is that it requires a plugin or bootstrapping to estimate T, which is required to compute the value of $\epsilon^*_{s,a}$.

Bellman-consistent Pessimism. Our algorithm is also not pessimistic under the weaker assumptions of Bellman-consistent pessimism. In this approach from Xie et al. (2021), the value of the starting state under any policy is estimated using the most pessimistic function in the space of value functions with low Bellman error. Then the policy that maximizes this pessimistic value estimate of the starting state is chosen. In our method, it is not required that the choice of Q_{reg} and the resulting value of ϵ^* that minimizes the MSE at the initial state be pessimistic.

Bellman-consistent pessimism applies to arbitrary function approximation and is well-suited for complex, higher-data settings, while our method provides simple approach appropriate for the low-data settings where discount regularization is generally beneficial. The main theoretical guarantee in Xie et al. (2021) requires searching over the policy space as well as evaluating each policy over all functions in the value function class with low Bellman error, which is not practical to implement. The version of the algorithm adapted for practical implementation avoids searching over the policy space, but still requires "access to a (regularized) loss minimization oracle over the value function class." They demonstrate that this can be efficiently implemented in linear function approximation, making Bellman-consistent pessimism a reasonable choice for settings where linear function approximation is used. Our method is more appropriate for small-data settings where function approximation is not appropriate, albeit without theoretical guarantees on the resulting performance.

6. Simulations

In this section, we empirically confirm our equivalence theorems and demonstrate the performance of our regularization methods in an offline setting.

6.1 Tabular Environments

We demonstrate our results on three common environments from the RL literature. The first comes from the initial work proposing discount regularization. We choose this environment to demonstrate the limitations of discount regularization even in an environment where it is known to be beneficial. We choose the other two because of their differences in structure, connectivity, and rewards to ensure that our results hold in diverse environments.

10-State Random Chain. The first environment is a distribution over MDPs and we sample one before generating each data set in the examples that follow. Jiang et al. (2015) empirically demonstrated the benefits of discount regularization on this randomly generated 10-state, 2-action MDP. For each state-action pair, 5 successor states are chosen at random to have nonzero transition probability. These probabilities are drawn independently from Uniform[0,1] and normalized to sum to one. The rewards are sampled independently from Uniform[0,1].

River Swim. This common tabular environment described in Osband et al. (2013) consists of six states and two actions, as illustrated in Figure 3. The agent can attempt to swim right "against the current" towards the larger reward, or swim left with probability 1 towards the smaller reward.

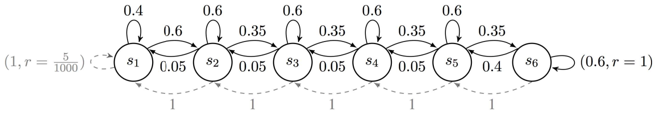


Figure 3: River Swim. Image from Osband et al. (2013).

GridWorld. The "GridWorld" environment is a modification of the environment with the same name from Amit et al. (2020). Like the 10-State Random Chain, it is a distribution over MDPs. The state space is a 4x4 grid where the agent's actions are left, right, up, or down. To construct the MDP, a probability p_s is drawn uniformly at random from [0,1] for each state s. With probability p_s, the agent moves in the desired direction if possible and otherwise remains in the same state. With probability $1 - p_s$, the agent transitions to a successor state that is randomly chosen for each state when the MDP is constructed. A single high-reward state is chosen at random to have a reward of 1, while all other states have rewards drawn uniformly at random from [-0.5,0.50].

6.2 Cancer Simulator Environment

We confirm our analysis in a larger, more realistic setting, using a cancer simulator developed by Ribba et al. (2012), as implemented by Gottesman et al. (2020). The simulator is based on data from patients with a type of tumor called low-grade gliomas (LGG). We use the version for chemotherapy drug TMZ. The structure of the model is based on 21 patients and parameters for the TMZ version are fit using data from 24 patients, with the remaining 96 held out for validation.

The state space consists of four dimensions: measurements for three different tumor tissue types and the drug concentration. We discretize the states by dividing each dimension into quartiles. The two actions represent whether or not to administer the chemotherapy drug TMZ at each time step, which represents one month. The reward at each time step is the reduction in total tumor size from the previous time step, minus a penalty for administering treatment at that time step. In the batch data, treatment at each time step is determined by a draw from the binomial distribution with treatment probability p. We compare regularization methods across a range of parameter choices: amount of stochasticity in the transition between states, magnitude of penalty to the reward for administering chemotherapy, noise in the starting state, and probability p of treatment in the batch data.

6.3 Model-Based Method Simulations

We showed analytically that planning under a reduced discount factor functions as a prior on the transition matrix with higher magnitude for state-action pairs with more transition observations. We then proposed a better way to regularize by deriving an explicit formula for a uniform prior that minimizes that transition matrix MSE locally for each state-action pair. Next we confirm our results empirically.

First we demonstrate that the equality in Thm. 1 holds. We then compare the performance of (1) discount regularization, (2) a uniform prior on T with equal magnitude for all state-action pairs, and (3) our state-action-specific regularization on the three simple tabular examples and the medical cancer simulator.

6.3.1 Procedure

To assess performance in each environment, we follow the procedure in Jiang et al. (2015). We repeatedly sample data sets from the true MDP. (A new MDP is sampled every time in the case of the 10-State Random Chain and GridWorld.) For each, we estimate the transition matrix from the data and assume the reward function is known. Then for a range of regularization strengths (ϵ or γ) we regularize the transition matrix separately using (1) discount regularization or (2) a uniform prior with constant magnitude across state-action pairs. We also regularize by (3) a uniform prior with state-action specific parameter. We then calculate the optimal policy. We compute the loss by taking the difference between the value of the true optimal policy in the true MDP and the value of the policy found in the estimated, regularized MDP, evaluated in the true MDP, and then averaging across states. The state-action-specific uniform prior is not dependent on a regularization parameter so we plot the single average loss value horizontally.

6.3.2 Empirical Demonstration of Equivalence Theorem: Discount Regularization and Uniform Prior on Transition Matrix Yield Identical Optimal Policies

First, we empirically confirm our result from Thm. 1. When the implied value of ϵ is the same for all state-action pairs, a uniform prior on T yields the same optimal policy as a planning discount factor of $\gamma(1-\epsilon)$. As per Eq. 3, we enforce equal ϵ across state-action pairs by sampling data sets with equal numbers of transition observations across state-action pairs. As demonstrated for the 10-State Random Chain environment in Fig. 4, loss is identical for both methods, as is expected for identical policies.

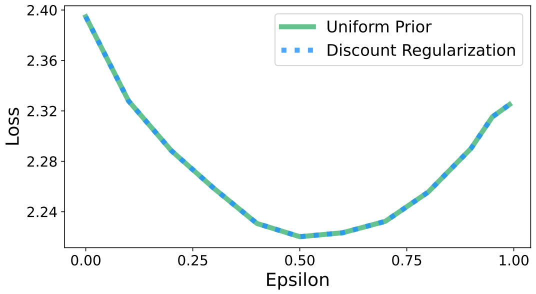
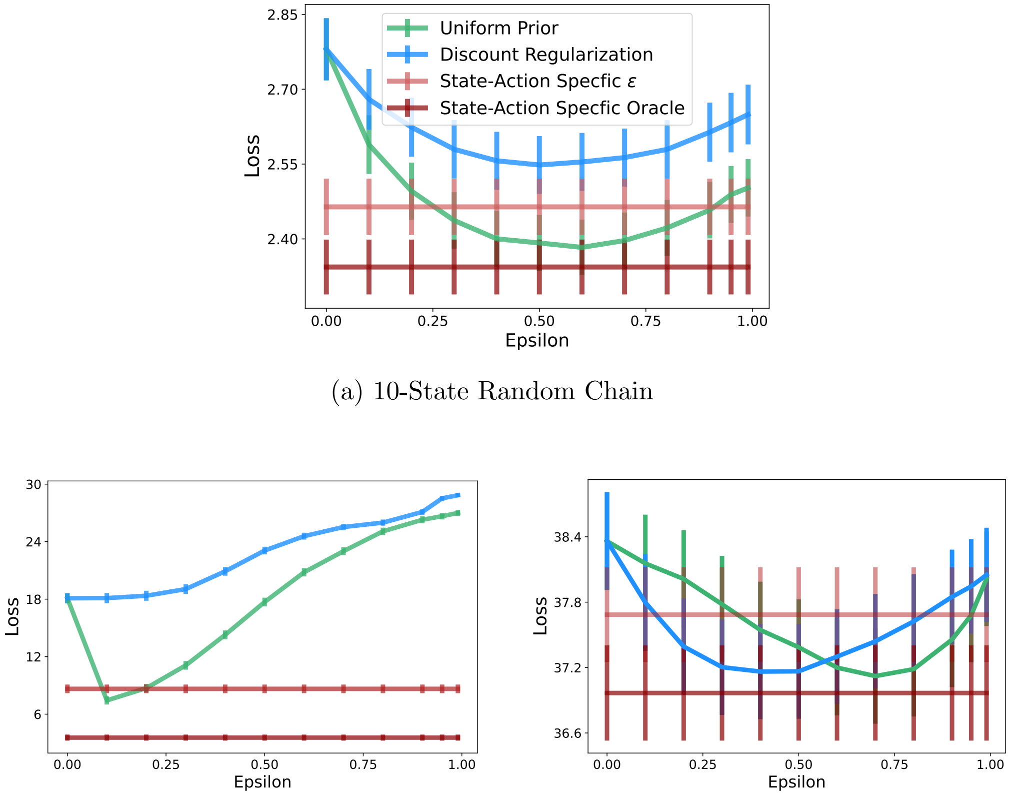


Figure 4: Discount regularization and a uniform prior on the transition matrix yield identical policies in model-based RL when transition count data are equal for all state-action pairs.

In the examples that follow, we relax the requirement of equal data across state-action pairs to compare methods under a more realistic data distribution.



(a) 10-State Random Chain

(b) River Swim

(c) GridWorld

Figure 5: A uniform prior on the transition matrix performs equally to or better than discount regularization in all three environments. A state-action-specific uniform prior performs close to or better than a uniform prior with global regularization parameter ϵ. "Oracle" represents state-action specific ϵ^* calculated using the true T. Calculated for 5,000 data sets.

6.3.3 Results: Problems with Discount Regularization

Discount Regularization performs poorly on data sets with uneven coverage across state-action pairs. In real-world conditions, it is unlikely that a data set will have equal numbers of transition observations across state-action pairs. In this case, recall that discount regularization functions as a prior with higher magnitude for state-action pairs with more data (Eq. 3). We compare this with a uniform prior on the transition matrix with equal magnitude for all state-action pairs. Fig. 5 shows the loss for each method across a range of values of ϵ (regularization strengths) for the three tabular environments. In these examples, the transition data is generated with starting state and action chosen uniformly at random, but transition counts are not enforced to be equal across state-action pairs. Even with transition data that is not heavily skewed away from uniform, the uniform prior

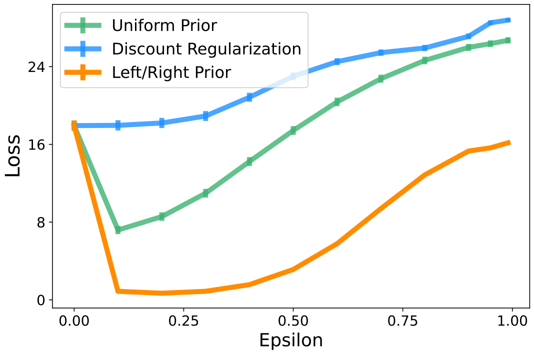


Figure 6: *River Swim Environment.* When we have knowledge about the environment, a prior chosen based on expert knowledge (the 'Left/Right Prior') can perform better.

with fixed magnitude generates policies that generally perform better (lower loss) in the true environment across a range of regularization strengths. Loss is significantly lower for a uniform prior compared to discount regularization in first two environments and similar in the case of the GridWorld environment.

Discount regularization performs poorly when the transition distribution differs greatly across states and/or actions. In addition to poor performance in skewed data sets, discount regularization does not perform well in cases where the implied prior, which has the same distribution for all (s, a), does not match the ground truth. For example, a domain expert may have knowledge that some state transitions are likely or others are impossible. Consider the case of River Swim. If a domain expert knows that Action 1 generally causes the agent to go left and Action 2 generally causes the agent to go right, we may choose a different prior on each action. For example, consider a prior on Action 1 that deterministically moves the agent left and the prior on Action 2 that deterministically moves the agent right. Fig. 6 compares the loss for this deterministic "left/right prior" with the other methods. Unsurprisingly, this hand-chosen prior results in lower loss than the methods which assume equal transition distributions for all states and actions.

6.3.4 Results: Our Method Provides Simple and Flexible State-Action-Specific Regularization

Performance depends not only on choosing an appropriate regularizer for the data set and environment, but also on setting the parameters correctly.

Our method avoids parameter tuning. Minimizing the transition matrix MSE equation with respect to regularization parameter $\epsilon_{i,j}$ yields an explicit formula for the parameter $\epsilon^*_{i,j}$, Eq. 7. This expression for $\epsilon^*_{i,j}$ depends inversely on the number of transition observations in the data, which allows for reduction in regularization with increased data.

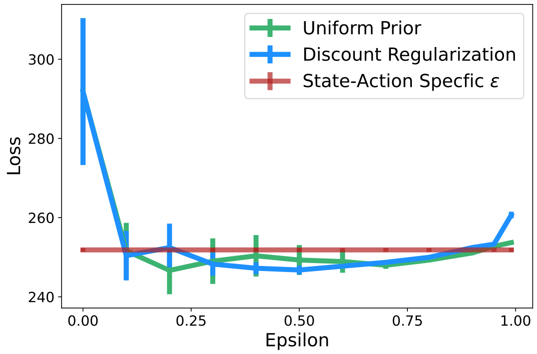


Figure 7: *Cancer Simulator.* State-action-specific regularization achieves near-minimum loss while avoiding the high loss resulting from incorrectly-set parameters.

The only quantity we lack is an estimate for T, which can be approximated by the MLE. Alternatively, we can model T from the data then sample from the posterior, choosing $\epsilon_{i,j}$ to minimize the MSE (Eq. 6) across the sampled estimates of T. Figures in this section reflect this approach. This is preferable to cross validation not only because it provides a simple, analytic form, but also because the situations in which regularization is beneficial generally involve few transition observations per state-action pair, resulting in insufficient amounts of data to divide into training and validation sets.

The gap between "state-action specific ϵ" loss and the "state-action specific oracle" loss in Fig. 5 is the difference in performance when using an estimate of T versus the true value. As expected, loss is higher when estimating T, however it still achieves loss near the minimum that can be achieved with a global regularization parameter without the risk of incurring the higher loss values that can result from an incorrectly-set global parameter.

The benefit of avoiding parameter tuning is also illustrated in the results from the cancer simulator in Fig. 7. Across variations in parameters, the two methods with global parameters performed similarly. However with both global methods, if ϵ is set incorrectly then the loss can be significantly higher. This makes state-action specific regularization particularly appealing, achieving loss near the minimum of all methods with the parameters set globally, but without tuning.

Our method remedies the issue of stronger regularization for state-action pairs with more data. Because the formula for $\epsilon^*_{i,j}$ is state-action-specific, it allows the flexibility to adjust the regularization amount separately across state-action pairs with different amounts of data and different transition distributions. This is particularly important as most real-world data sets have uneven distributions, and enforcing equal regularization across state-action pairs in that case impedes performance.

Returning to Fig. 5, we demonstrate that our state-action-specific regularization reduces loss without parameter tuning. The horizontal line for "State-action-specific ϵ" represents the loss when regularization parameter $\epsilon_{i,j}$ is set separately for each state-action pair. A state-action-specific regularization parameter yields loss that outperforms discount regularization and is close to or outperforms a uniform prior of constant magnitude.

6.4 Model-Free Method Simulations

Discount regularization is used in both model-based and model-free settings, so we extend our simulations to a model-free setting. As in the model-based setting, we first demonstrate the equivalence theorem, then compare the performance of discount regularization with our state-action specific methodology. We then discuss the reasons that model-based outperforms model-free in the low-data settings that we consider.

6.4.1 Procedure

We follow an analogous procedure as in the model-based case, adapted to the model-free RL method FQI. As above, we repeatedly sample data from the true MDP. For each data set, for a range of regularization strengths (ϵ or γ), we run FQI using either (1) a lower discount factor or (2) a weighted average Q-function update reflecting Eq. 4. The reward function is assumed known. Loss is the difference between the value of the true optimal policy and the value of the policy found with the regularized version of FQI, both evaluated in the true MDP, averaged across states. The state-action-specific method is not dependent on a regularization parameter so we plot the single average loss value horizontally in Fig. 9.

Note that unlike the model-based setting where any planning discount factor γ_p maps to a prior on the transition matrix, there is no simple mapping between the planning discount factor and an implied prior in the model-free case. Consequently our simulations lack an equivalent comparison between discount regularization and a fixed prior of equal magnitude for all state-action pairs.

6.4.2 Empirical Demonstration of Equivalence Theorem: Discount Regularization and Weighted average Q-function Yield Identical Optimal Policies

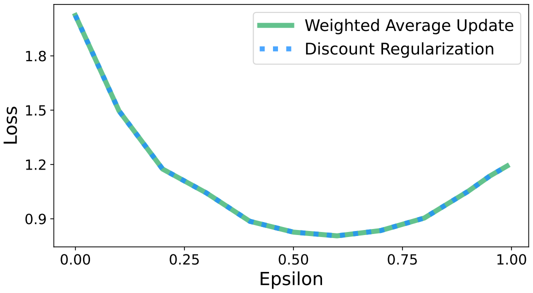


Figure 8: Discount regularization and a weighted average Q-function produce identical policies in model-free RL, resulting in equal loss for both methods.

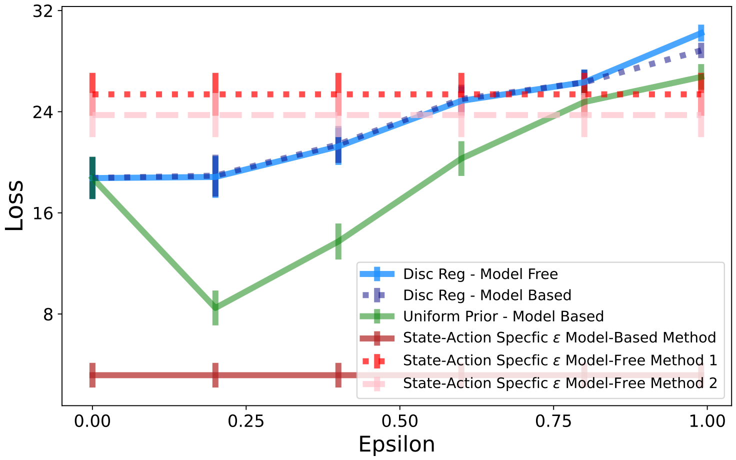


Figure 9: *River Swim Environment.* Comparison of model-free and model-based regularization methods. State-action-specific methods show loss without estimation error, using true value of T or Q.

As we did in the model-based case, we empirically confirm the model-free equivalence result from Thm. 2. When regularization parameter $\epsilon_{i,j}$ is the same for all state-action pairs, discount regularization and regularization using the weighted average Q-function in Eq. 4 yield the same optimal policy. Fig. 8 demonstrates that loss is identical for both methods, as is expected for identical policies.

6.4.3 Results: Model-free state-action-specific regularization underperforms model-based

We evaluate the loss from policies using discount regularization and using our state-action-specific regularization parameter $\epsilon^*_{i,j}$ for a range of environments and data set sizes. We found that our model-free method did not perform well, particularly in comparison to model-based methods. One source of loss in the model-free method is estimation error from using the current estimate of Q in each step of FQI. To assess the best possible performance of the method, without this estimation error, we computed $\epsilon^*_{i,j}$ using the true values of Q^{π^*}. The loss for each of our methods without estimation error is illustrated in Figure 9, with results for additional data set sizes and environments are in Appendix E. Although, after removing the estimation error, our method resulted in lower loss compared to not regularizing in the 10-State Random Chain and GridWorld environments, loss was still higher than than of model-based methods.

Another cause of underperformance stems from the choice to regularize Q instead of T. The model-based method chooses the value of $\epsilon_{i,j}$ that minimizes the error between the regularized estimate of $T(s_i, a_j, \cdot)$ and its true value. The values for both are on the same scale: ranging from 0 to 1 and summing to 1. On the other hand, the model-free method chooses the value of $\epsilon_{i,j}$ that minimizes the error between the regularized estimate of $Q(s_i, a_j)$ and its true value, which can differ significantly. These differences push the values of $\epsilon^*_{i,j}$ towards 0 or 1. For example, the estimated Q-values could all be higher than the true Q-values. In this case, the values of $\epsilon_{i,j}$ that minimizes the MSE in $\hat{Q}(s_i, a_j)$ are all either 1 (for estimated $Q(s_i, a_j)$ above than average estimate) or 0 (for estimated $Q(s_i, a_j)$ below the average). Our goal in regularization is to obtain a policy that performs well in the true environment, for which we must set epsilon for all (s_i, a_j) together in a way that minimizes the error in *relative* values. Minimizing the absolute error for each $Q(s_i, a_j)$ does not achieve this goal.

7. Extension to Continuous States

Theorems 1 and 2 and our resulting regularization methods apply to any number of increasingly smaller discrete states. It stands to reason, then, that we can extend our methodology to a continuous state space. We demonstrate how our state-action-specific regularization methodology can be successfully applied to a continuous state space below.

7.1 Methodology

We first describe how our continuous-state approach extends the ideas from the discrete-state setting, and then apply these ideas to a continuous setting where the transition function is modeled using kernel regression.

7.1.1 Connection to Discrete-State Methodology

In discrete-state environments, discount regularization is commonly used with certainty-equivalence RL. Certainty-equivalence RL uses a single estimate of the transition distribution, such as the MLE, rather than modeling the uncertainty in the transition distribution. This estimate may be regularized to avoid overfitting. We showed that setting the regularization amount separately for each state-action pair can result in better-performing policies (lower loss). We extend this concept to a one-dimensional continuous state space by modeling T with a single estimate of next state given state and action, rather than using a more complex transition model that reflects the uncertainty over next state, and then regularizing this estimate before computing the optimal policy.

As discussed above, Theorems 1 and 2 imply that discount regularization pushes the values of all state-action pairs closer together. In our state-action-specific regularization method, we control the amount that the value of each state-action pair moves towards the mean separately. The same concept applies in a continuous state space. We demonstrate one way of applying these concepts to regularization with a continuous state space below.

7.1.2 Kernel Regression Transition Model Method

To apply our insights from a tabular state space to a continuous state space, we take the example of modeling the transition dynamics using kernel regression and then computing the optimal policy by fitted value iteration (FVI, Szepesvári and Munos 2005). We first model the expected next state (given state and action) $s' = E[T(s, a)]$ using a kernel regression separately for each action (specifically the Nadaraya-Watson Kernel estimator; Nadaraya 1964, Watson 1964). As in the tabular case, we regularize by planning as if the agent transitions according to any chosen distribution that is the same for all state-action pairs with probability ϵ. In that case, our regularized estimator of the expected next state given state and action is

$$\hat{T}_\epsilon(s, a_j) = (1-\epsilon) \underbrace{\frac{\sum_{d=1}^{D} K_h(s - s_d) s'_d}{\sum_{d=1}^{D} K_h(s - s_d)}}_{\text{kernel regression estimate}} + \underbrace{\epsilon T_{reg}}_{\substack{\text{mean of regularization} \\ \text{transition distribution}}}, \qquad (10)$$

where $\{s_d, s'_d\}_{d=1}^{D}$ are the state and next state observations in the data for action a_j, K_h is a kernel function with bandwidth h and T_{reg} is the mean of the chosen regularization transition distribution.

We select $\epsilon^*(s, a_j)$, the value of ϵ that minimizes the MSE of $\hat{T}_\epsilon(s, a_j)$ by using the approximations of bias and variance of the Nadaraya-Watson Kernel estimator implied by the kernel regression confidence bounds presented in Wasserman (2004, p. 323). This yields the expression for $\epsilon^*(s, a_j)$ below, derived in Appendix F,

$$\epsilon^*(s, a_k) = \frac{\hat{se}^2(s, a_k)}{(T_{reg} - \hat{T}_{NW}(s, a_k))^2 + \hat{se}^2(s, a_k)}, \qquad (11)$$

where:

- $\hat{se}(s, a_k) = \hat{\sigma}(s, a_k) \sqrt{\sum_{i=1}^{D} \left[\frac{K_h(s - s_i)}{\sum_{j=1}^{D} K_h(s - s_j)} \right]^2}$
- $\hat{\sigma}^2 = \frac{1}{2(n-1)} \sum_{d=1}^{D-1} (s'_{d+1} - s'_d)^2$ where the data is ordered by the values of $\{s\}_d$,
- $\hat{T}_{NW}$ is the Nadaraya-Watson kernel estimator for the expected next state given current state and action, $\hat{T}_{NW} = \frac{\sum_{d=1}^{D} K_h(s - s_d) s'_d}{\sum_{d=1}^{D} K_h(s - s_d)}$,
- and values above are computed over the D data tuples $\{s_d, a_k, s'_d\}_{d=1}^{D}$ separately for each given discrete action a_k.

We use this state-action-specific regularization parameter $\epsilon^*(s, a_k)$ in FVI. The procedure is detailed in Algorithm 2. We calculate $\epsilon^*(s, a_k)$ before all iterations of FVI. Then we use a weighted average Bellman update like in the tabular model-free case: using $\epsilon^*(s, a_k)$ to average the value of the next state as predicted by kernel regression and the value of the next state predicted by our fixed regularization transition distribution.

By modeling only the expected next state given state and action rather than the full distribution over next state, Algorithm 2 is more efficient than sampling-based FVI where the next state is repeatedly sampled from the model for T and $Q(s_n, a_m)$ is estimated by averaging over samples. Note that line 9 of the algorithm assumes the uniform distribution over states as the regularization distribution, but this can be modified to reflect any transition distribution as appropriate for the environment.

7.2 Continuous-State Simulation

We demonstrate the regularization method described above on a simple continuous-state environment. The environment, simulation procedure and results are described below.

Algorithm 2 FVI with State-Action-Specific Regularization

```
1: Sample $N$ states $\{s_n\}_{n-1}^N$ from state space
2: Initialize value function model $V_\theta(s)$
3: Learn transition model for next state given state and action $s'|s, a = \hat{T}(s, a)$ using batch
   data and kernel regression.
4: Predict next state $s'_{n,k}|s_n, a_k$ for all sampled states $s_n$ and all actions $a_k \in A$ using
   transition model from line 3.
5: Calculate $\epsilon^*(s_n, a_k)$ for all sampled states and all actions (Eq. 11).
6: for i in num_iters do
7:     for each sample state $\{s_n\}_{n=1}^N$ do
8:         for each action $a_k \in A$ do
9:             $Q(s_n, a_k) = R(s_n, a_k) + \gamma(1 - \epsilon^*(s_n, a_k))V_\theta(s'_{n,k}) + \gamma\epsilon^*(s_n, a_k)\frac{1}{N}\sum_{n'=1}^N V_\theta(s'_{n'})$
10:        end for
11:        $v_n = \max_k Q(s_n, a_k)$
12:    end for
13:    Update $V_\theta(s)$ using supervised learning and $\{s_n, v_n\}_{n=1}^N$
14: end for
```

7.2.1 Environment

Continuous River Swim. We demonstrate the performance of Algorithm 2 on a continuous version of the River Swim environment. We take $s \in [0, 1]$ to be the one-dimensional continuous state space. We define a continuous reward function that produces a smaller positive reward near the lower end of the state space and a larger positive reward at the high end. As in the tabular version, there are two discrete actions. Action 0 moves the agent "against the current" (i.e. with more stochasticity) towards the higher reward region and action 1 moves the agent "with the current" (with less stochasticity) towards the lower reward region. Stochasticity is introduced via random uniform noise. See Appendix G for more details.

7.2.2 Procedure

Like in the tabular cases, we repeatedly sample data sets of tuples $\{s, a, s'\}$ from the true MDP. We compute the optimal policy using Algorithm 2 with either discount regularization (equivalent to a fixed value of ϵ for all states and actions) or using the state-action specific $\epsilon^*(s, a_k)$ in Eq. 11. The loss is the difference between the value of the true optimal policy and the value of the policy found using the regularized, estimated transition function, each evaluated in the true environment and averaged across sampled states. We use randomized decision trees (scikit-learn ExtraTreesRegressor; Pedregosa et al., 2011) to model value function $V(s)$.

7.2.3 Results

Figure 10 displays the average loss across data sets for a range of values of ϵ for discount regularization, as well as loss for the state-action specific $\epsilon^*(s, a_k)$ (single loss value plotted horizontally). For this environment, the state-action specific regularization method results

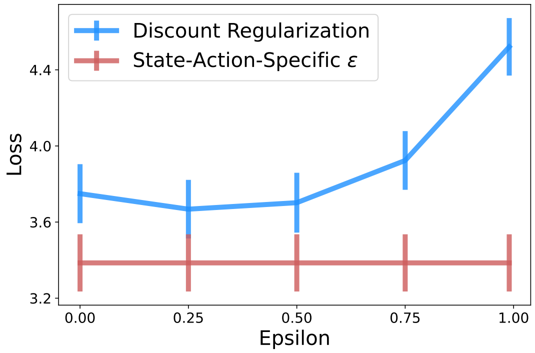


Figure 10: State-action specific regularization results in lower loss than discount regularization in the continuous-state River Swim environment.

in lower loss than any fixed global value of ϵ. This includes outperforming the unregularized version of FVI ($\epsilon = 0$). Our state-action-specific method achieves these results with no parameter tuning or cross-validation, only a single computation for any (s, a).

This method allows us to use a simple transition model with lower loss, however we note that, as discussed in the tabular case, the performance depends on the underlying transition distribution matching the chosen regularization distribution. This choice must be made carefully based on knowledge of the environment. Also, note that because our algorithm regularizes the Bellman equation directly (not the transition matrix), discount regularization and fixed ϵ weighted average generate identical policies and hence produce the same loss.

8. Discussion

Comparison of Model-Based and Model-Free Methods. The regularization methods that we introduce in this paper improve upon the performance of discount regularization, allowing us to use simple RL methods with limited data. Simulations revealed that our model-based methods were more successful than the model-free methods in low-data settings where regularization is needed. This is in agreement with the idea that model-based methods perform better in low-data settings due to their sample efficiency (Yu, 2018).

Extension to Online Methods. The connection between discount regularization and a weighted average transition matrix arises most obviously in the batch setting, however our methods to mitigate the pitfalls of discount regularization can be applied to online methods as well. Because the expression for $\epsilon^*_{i,j}$ decreases towards 0 with increasing number of transitions observed, it provides a way to decrease regularization as data is collected online. For example, we can modify the Q-learning algorithm to incorporate a weighted average update rule at each step. Please see Appendix G.1 for algorithm details and proof of concept.

Limitations. Our model-based method is based on learning a transition model with low MSE, which does not guarantee a good policy. In other words, it is possible to learn a good policy from a "bad" transition model and vice versa, in particular because certain errors in the model may affect the policy more than others. However, this method works well in comparison to our model-free method which directly minimizes error in Q demonstrating that minimizing the error in T rather than Q is a reasonable approach.

9. Conclusion

Discount regularization is a commonly used technique for dealing with noisy and sparse data. Although practitioners believe that they are simply ignoring delayed effects, we revealed deeper connections to other methods of regularization. In the model-based context, through a simple reframing of discount regularization as a weighted average transition matrix, we showed that it acts like a prior on the transition matrix that has the same distribution for all states and actions. Applying the same reframing in a model-free context revealed a connection between discount regularization and regularization methods that penalize that value function.

We showed that, problematically, discount regularization results in stronger regularization for state-action pairs with more data. To remedy the issue, we used the weighted average form to derive an explicit formula for the regularization parameter that is calculated locally for each state-action pair rather than globally. We demonstrated that this approach results in lower loss without parameter tuning in model-based RL, both with discrete or continuous states. In model-free RL, this approach was not effective, consistent with the understanding that model-based methods perform better in low-data settings.

Acknowledgments

Research reported in this work was supported by the following grants:

NIH/NIDA P50DA054039, NIH/NIBIB and OD P41EB028242;

NIH award 5R01MH123804-02.

This material is based upon work supported by the National Science Foundation under Grant No. IIS-2007076. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Susan Murphy holds concurrent appointments at Harvard University and as an Amazon Scholar. This paper describes work performed at Harvard University and is not associated with Amazon.

Appendix A. Unified Form Derivations

A.1 Dirichlet Prior Derivation

Assume prior $T_{prior}(s_i, a_j, \cdot) = \text{Dirichlet}(\langle \alpha_{i,j,1}, ..., \alpha_{i,j,N_s} \rangle)$ on transition matrix $T(s_i, a_j)$ and let $\langle c_{i,j,1}, ..., c_{i,j,N_s} \rangle$ be the transition count data observed from state s_i to states 1 through N_s under action a_j. The posterior of $T(s_i, a_k, \cdot)$ follows a Dirichlet distribution with parameter $\langle c_{i,j,1} + \alpha_{i,j,1}, ..., c_{i,j,N_s} + \alpha_{i,j,N_s} \rangle$ and the posterior mean is:

$$
\begin{aligned}
T_{\text{post mean}}(s_i, a_j, \cdot) &= \langle \frac{c_{i,j,1} + \alpha_{i,j,1}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}}, ..., \frac{c_{i,j,N_s} + \alpha_{i,j,N_s}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \rangle \\
&= \langle \frac{c_{i,j,1}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}}, ..., \frac{c_{i,j,N_s}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \rangle \\
&+ \langle \frac{\alpha_{i,j,1}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}}, ..., \frac{\alpha_{i,j,N_s}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \rangle \\
&= \frac{\sum\limits_{k=1}^{N_s} c_{i,j,k}}{\sum\limits_{k=1}^{N_s} c_{i,j,k}} \langle \frac{c_{i,j,1}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}}, ..., \frac{c_{i,j,N_s}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \rangle \\
&+ \frac{\sum\limits_{k=1}^{N_s} \alpha_{i,j,k}}{\sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \langle \frac{\alpha_{i,j,1}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}}, ..., \frac{\alpha_{i,j,N_s}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \rangle \\
&= \frac{\sum\limits_{k=1}^{N_s} c_{i,j,k}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \langle \frac{c_{i,j,1}}{\sum\limits_{k=1}^{N_s} c_{i,j,k}}, ..., \frac{c_{i,j,N_s}}{\sum\limits_{k=1}^{N_s} c_{i,j,k}} \rangle \\
&+ \frac{\sum\limits_{k=1}^{N_s} \alpha_{i,j,k}}{\sum\limits_{k=1}^{N_s} c_{i,j,k} + \sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \langle \frac{\alpha_{i,j,1}}{\sum\limits_{k=1}^{N_s} \alpha_{i,j,k}}, ..., \frac{\alpha_{i,j,N_s}}{\sum\limits_{k=1}^{N_s} \alpha_{i,j,k}} \rangle
\end{aligned}
$$

Let $\hat{T}_{MLE}(s_i, a_j, \cdot)$ be the MLE of $T(s_i, a_j, \cdot)$. Then,

$$
\hat{T}_{MLE}(s_i, a_j, \cdot) = \langle \frac{c_{i,j,1}}{\sum\limits_{k=1}^{N_s} c_{i,j,k}}, ..., \frac{c_{i,j,N_s}}{\sum\limits_{k=1}^{N_s} c_{i,j,k}} \rangle.
$$

Let $T_{\text{prior mean}}(s_i, a_j, \cdot)$ be the transition matrix implied by the prior for state s_i and action a_j, i.e. $T_{\text{prior mean}}(s_i, a_j, \cdot) = \langle \frac{\alpha_{i,j,1}}{\sum_{k=1}^{N_s} \alpha_{i,j,k}}, ..., \frac{\alpha_{i,j,N_s}}{\sum_{k=1}^{N_s} \alpha_{i,j,k}} \rangle$. Then we can write $T_{\text{post mean}}(s_i, a_j, \cdot)$

as follows.

$$T_{\text{post mean}}(s_i, a_j, \cdot) = \frac{\sum_{k=1}^{N_s} c_{i,j,k}}{\sum_{k=1}^{N_s} c_{i,j,k} + \sum_{k=1}^{N_s} \alpha_{i,j,k}} \hat{T}_{MLE}(s_i, a_j, \cdot) + \frac{\sum_{k=1}^{N_s} \alpha_{i,j,k}}{\sum_{k=1}^{N_s} c_{i,j,k} + \sum_{k=1}^{N_s} \alpha_{i,j,k}} T_{\text{prior mean}}(s_i, a_j, \cdot)$$

Let $\epsilon = \frac{\sum_{k=1}^{N_s} \alpha_{i,j,k}}{\sum_{k=1}^{N_s} c_{i,j,k} + \sum_{k=1}^{N_s} \alpha_{i,j,k}}$. Then we have:

$$T_{\text{post mean}}(s_i, a_j, \cdot) = (1-\epsilon)\hat{T}_{MLE}(s_i, a_j, \cdot) + \epsilon T_{\text{prior mean}}(s_i, a_j, \cdot)$$

A.2 Discount Regularization

Consider the matrix form of the Bellman equation, using γ_p, the lower value of the discount factor used during planning for regularization: $V = R + \gamma_p TV$. By the steps below, we write the product $\gamma_p T$ from the Bellman equation as the product of true discount factor γ and a weighted average matrix.

$$\gamma_p T = [\gamma - (\gamma - \gamma_p)]T \qquad \text{(Add and subtract } \gamma\text{)}$$

$$\gamma_p T = \gamma(1 - \frac{(\gamma - \gamma_p)}{\gamma})T \qquad \text{(Pull out a factor of } \gamma\text{.)}$$

Let T_{zeros} be an appropriately sized matrix of zeros. Adding γT_{zeros} to the right hand side does not change the equality.

$$\gamma_p T = \gamma[(1 - \frac{\gamma - \gamma_p}{\gamma})T + T_{zeros}]$$

Multiply the T_{zeros} term inside the parentheses by $\frac{\gamma - \gamma_p}{\gamma}$. T_{zeros} is all zeros so a multiplier does not affect the equality.

$$\gamma_p T = \gamma[(1 - \frac{\gamma - \gamma_p}{\gamma})T + (\frac{\gamma - \gamma_p}{\gamma})T_{zeros}]$$

Let $\epsilon = \frac{\gamma - \gamma_p}{\gamma}$.

$$\gamma_p T = \gamma[(1-\epsilon)T_{true} + \epsilon T_{zeros}]$$

We have replaced the product of the planning discount factor and the true transition matrix with the product of the true discount factor and a weighted average of the transition matrix and a matrix of zeros. To put this in our framework, consider regularizing the MLE transition matrix for state-action pair (s, a) via discount regularization, using planning discount factor γ_p. Using the proof in this section, our regularized estimated transition matrix $\hat{T}(s, a)$, is:

$$\hat{T}(s, a, \cdot) = (1 - \epsilon)\hat{T}_{MLE}(s, a, \cdot) + \epsilon T_{zeros}$$

where $\epsilon = \frac{\gamma - \gamma_p}{\gamma}$.

A.3 Dirichlet Prior Implied by Discount Regularization

The proof of equivalence in Thm. 1 demonstrates that, for a given value of ϵ, averaging the transition matrix with either a matrix of the discrete uniform distribution or with the matrix of zeros yields the same policy. Since discount regularization applies the same ϵ to every state-action pair, the two methods are only exactly equivalent when the posterior transition matrix under a uniform prior has the same implied value of ϵ for all state-action pairs, i.e. $\epsilon = \frac{\sum_{k=1}^{N_s} \alpha_{i,j,k}}{\sum_{k=1}^{N_s} c_{i,j,k} + \sum_{k=1}^{N_s} \alpha_{i,j,k}}$ is the same for all state-action pairs, where $c_{i,j,k}$ and $\alpha_{i,j,k}$ are the transition count and prior from state i to state k under action j.

We can use the equivalence of the two methods for the same value of ϵ to solve for the magnitude of prior that is implied by a given discount factor in discount regularization. This is particularly interesting when $\sum_{k=1}^{N_s} c_{i,j,k}$ is unequal across state-action pairs (s_i, a_j). In this case, we can use the equivalence of discount regularization and the uniform prior to compute the different priors implied by discount regularization across state-action pairs.

Below we refer to $\sum_{k=1}^{N_s} c_{i,j,k}$ as $\sum c_k$ and $\sum_{k=1}^{N_s} \alpha_{i,j,k}$ as $\sum \alpha_k$ for brevity. Since we know both methods produce the same optimal policy for equal values of ϵ, we set the formulas for ϵ under each method equal to one another. We then solve for $\sum \alpha_k$ to get the magnitude of prior that is implied by a given planning discount factor γ_p.

$$\begin{aligned}
\frac{\sum \alpha_k}{\sum \alpha_k + \sum c_k} &= \frac{\gamma - \gamma_p}{\gamma} \\
\gamma \sum \alpha_k &= (\sum \alpha_k + \sum c_k)(\gamma - \gamma_p) \\
\gamma \sum \alpha_k &= \gamma \sum \alpha_k - \gamma_p \sum \alpha_k + \gamma \sum c_k - \gamma_p \sum c_k \\
\gamma_p \sum \alpha_k &= \sum c_k(\gamma - \gamma_p) \\
\sum \alpha_k &= \sum c_k \frac{\gamma - \gamma_p}{\gamma_p}
\end{aligned}$$

For the uniform distribution, all α_k for a given state are the same, so substitute $\sum \alpha_k = N\alpha_k$.

$$\begin{aligned}
N\alpha_k &= \sum c_k \frac{\gamma - \gamma_p}{\gamma_p} \\
\alpha_k &= \frac{\sum c_k}{N_s} \frac{\gamma - \gamma_p}{\gamma_p}
\end{aligned}$$

So discount regularization functions like the Dirichlet prior:

$$T_{\text{prior}}(s_i, a_j, \cdot) \sim \text{Dirichlet}(\frac{\sum c_k}{N_s}\frac{\gamma - \gamma_p}{\gamma_p}, ..., \frac{\sum c_k}{N_s}\frac{\gamma - \gamma_p}{\gamma_p}) \tag{12}$$

where again $\sum c_k$ is the total number of transitions observed in the data starting at state s_i under action a_j.

Appendix B. Proof of Thm. 1 Step 1

Suppose $Q^*_1(s, a)$ is the unique solution to:

$$(a) \quad Q^*_1(s, a) = R(s, a) + c + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q^*_1(s', a')$$

and $Q^*_2(s, a)$ is the unique solution to:

$$(b) \quad Q^*_2(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q^*_s(s', a')$$

.

First note that $Q^*_1(s, a) = \max_\pi E_{T,\pi}[\sum_{t=0}^{\infty} \gamma^t (R(S_t, A_t) + c)|S_0 = s, A_0 = a]$ is the unique solution to Eq. (a) and $Q^*_2(s, a) = \max_\pi E_{T,\pi}[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t)|S_0 = s, A_0 = a]$ is the unique solution to Eq. (b).

But $Q^*_1(s, a) = \max_\pi E_{T,\pi}[\sum_{t=0}^{\infty} \gamma^t R(S_t, A_t)|S_0 = s, A_0 = a] + \sum_{t=0}^{\infty} \gamma^t c = Q^*_2(s, a) + \frac{c}{1-\gamma}$. Thus Q^*_1 and Q^*_2 have the same argmax and correspond to the same optimal policy, $\pi^*_1 = \pi^*_2$.

Appendix C. Uniform Prior MSE Calcuation

Let $c_{i,j} = \sum_{k=1}^{N_s} c_{i,j,k}$ be the transition observation count for state-action pair (s_i, a_j) in the data. Let $T(s_i, a_j, \cdot)$ be the transition probability distribution under action a_j starting at state s_i. N_s is the number of states in the MDP.

$$\text{MSE}(\hat{T}(s_i, a_j, \cdot)) = \sum_{k=1}^{N_s} \Big(\text{Variance}(\hat{T}(s_i, a_j, s_k)) + \text{Bias}^2(\hat{T}(s_i, a_j, s_k))\Big)$$

$$\begin{aligned}
\text{Variance}(\hat{T}_{\text{unif}}(s_i,a_j,s_k)) &= \text{Variance}\left((1-\epsilon_{i,j})\hat{T}_{\text{MLE}}(s_i,a_j,s_k)+\epsilon_{i,j}\frac{1}{N_s}\right)\\
&= \text{Variance}\left((1-\epsilon_{i,j})\hat{T}_{\text{MLE}}(s_i,a_j,s_k)\right)+\text{Variance}\left(\epsilon_{i,j}\frac{1}{N_s}\right)\\
&= (1-\epsilon_{i,j})^2\text{Variance}(\hat{T}_{\text{MLE}}(s_i,a_j,s_k))\\
&= (1-\epsilon_{i,j})^2\frac{1}{c_{i.j}}T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))\\
\text{Bias}(\hat{T}_{\text{unif}}(s_i,a_j,s_k)) &= \mathbb{E}\left[\hat{T}_{\text{unif}}(s_i,a_j,s_k)-T(s_i,a_j,s_k)\right]\\
&= \mathbb{E}\left[(1-\epsilon_{i,j})\hat{T}_{\text{MLE}}(s_i,a_j,s_k)+\epsilon_{i,j}\frac{1}{N_s}-T(s_i,a_j,s_k)\right]\\
&= (1-\epsilon_{i,j})T(s_i,a_j,s_k)+\epsilon_{i,j}\frac{1}{N_s}-T(s_i,a_j,s_k)\\
&= \epsilon_{i,j}\left(\frac{1}{N_s}-T(s_i,a_j,s_k)\right)
\end{aligned}$$

$$\begin{aligned}
\text{MSE}(\hat{T}_{\text{unif}}(s_i,a_j,\cdot)) = \sum_{k=1}^{N_s}&[(1-\epsilon_{i,j})^2\frac{1}{c_{i.j}}T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))\\
&+\epsilon_{i,j}^2(\frac{1}{N_s}-T(s_i,a_j,s_k))^2]
\end{aligned}$$

To solve for the value of $\epsilon_{i,j}$ that minimizes the MSE, set the first derivative equal to 0.

$$\begin{aligned}
\frac{\partial \text{MSE}}{\epsilon_{i,j}} = \sum_{k=1}^{N_s} &-2(1-\epsilon_{i,j})\frac{1}{c_{i.j}}T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))\\
&+2\epsilon_{i,j}\left(\frac{1}{N_s}-T(s_i,a_j,s_k)\right)^2=0
\end{aligned}$$

$$\begin{aligned}
\sum_{j=1}^{N_s}&-2\frac{1}{c_{i.j}}T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))+2\epsilon_{i,j}\frac{1}{c_{i.j}}T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))\\
&+2\epsilon_{i,j}\left(\frac{1}{N_s}-T(s_i,a_j,s_k)\right)^2=0
\end{aligned}$$

$$\begin{aligned}
\sum_{k=1}^{N_s}&\epsilon_{i,j}\left[\frac{1}{c_{i.j}}T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))+\left(\frac{1}{N_s}-T(s_i,a_j,s_k)\right)^2\right]\\
&=\sum_{k=1}^{N_s}\frac{1}{c_{i.j}}T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))
\end{aligned}$$

$$
\begin{aligned}
\epsilon_{i,j} &= \frac{\sum\limits_{k=1}^{N_s} \frac{1}{c_{i.j}} T(s_i, a_j, s_k)(1 - T(s_i, a_j, s_k))}{\sum\limits_{k=1}^{N_s} \frac{1}{c_{i,j}} T(s_i, a_j, s_k)(1 - T(s_i, a_j, s_k)) + \left(\frac{1}{N_s} - T(s_i, a_j, s_k)\right)^2} \\
&= \frac{\sum\limits_{k=1}^{N_s} T(s_i, a_j, s_k)(1 - T(s_i, a_j, s_k))}{\sum\limits_{k=1}^{N_k} T(s_i, a_j, s_k)(1 - T(s_i, a_j, s_k)) + c_{i,j}\left(\frac{1}{N_s} - T(s_i, a_j, s_k)\right)^2} \\
&= \frac{\sum\limits_{k=1}^{N_s} [T(s_i, a_j, s_k)(1 - T(s_i, a_j, s_k))] / \sum_{k=1}^{N_s} [(\frac{1}{N_s} - T(s_i, a_j, s_k))^2]}{\sum\limits_{k=1}^{N_s} [T(s_i, a_j, s_k)(1 - T(s_i, a_j, s_k))] / \sum\limits_{k=1}^{N_s} [(\frac{1}{N_s} - T(s_i, a_j, s_k))^2] + c_{i,j}}
\end{aligned}
$$

Appendix D. Model Free State-action-specific Regularization Calculations

We calculate mean squared error in value $Q(s, a)$ for for given state-action pair (s, a) with fixed policy π—then solve for the value of ϵ that minimizes error.

D.1 Definitions

We define the following variables:

- N_s: number of states
- N_a: number of actions
- C: Count data matrix, dimension $N_sN_a \times N_s$ (one row for every (s,a), one column for every s')
- Q^π: Matrix of state-action values, $N_sN_a \times 1$ (one row for every (s,a))
- $\mathbf{1}$: A vector of ones of appropriate size
- Π: matrix mapping Q^π to V^π based on policy π, dimension $N_sN_a \times N_s$. There is a column for each state and a row for each state-action pair. For each state column, the row corresponding to the state-action pair of the optimal action gets a 1, all others 0.
- Diag$(\mathcal{E})$: $N_sN_a \times N_sN_a$ matrix with $\epsilon(s, a)$ across main diagonal, 0s elsewhere.
- v^π: $N_s \times 1$ vector of state values under policy π, equal to $\Pi^T Q^\pi$
- $[\Pi^T Q^\pi]_{sq}$: $N_s \times 1$ elementwise square of $\Pi^T Q^\pi$, the value under policy π

D.2 State-Action-Specific Regularization Parameter Calculation Method 1

The first way we calculate the value of $\epsilon^*(s, a)$ to use at each iteration of our modified fitted Q-iteration (Algo 1) is to calculate the sum of squared errors across all transition data (s, a, s') in the batch data set, then solve for the value of ϵ that minimizes the SSE.

$$
\begin{aligned}
SSE = &\sum_{i,j}\sum_{k=1}^{N_s}\sum_{s'|(s_i,a_j)\text{in data}} \mathbb{1}\{s'_{\text{data}=s_k}\}\big(R(s_i,a_j)+\gamma(1-\epsilon_{i,j})Q^{\pi}(s_k,\pi(s_k))\\
&+\gamma\epsilon_{i,j}\frac{1}{N_s}\sum_{k'=1}^{N_s} Q^{\pi}(s_{k'},\pi(s_{k'}))-Q^{\pi}(s_i,a_j)\big)^2\\
=&\sum_{i,j}\sum_{k=1}^{N_s} c_{i,j,k}\big(R(s_i,a_j)+\gamma(1-\epsilon_{i,j})Q^{\pi}(s_k,\pi(s_k))\\
&+\gamma\epsilon_{i,j}\frac{1}{N_s}\sum_{k'=1}^{N_s} Q^{\pi}(s_{k'},\pi(s_{k'}))-Q^{\pi}(s_i,a_j)\big)^2
\end{aligned}
$$

Set the derivative with respect to $\epsilon_{i,j}$ for one specific (i,j) equal to 0. Call the state-action pair that the differentiation is with respect to (i^*, j^*).

$$
\begin{aligned}
\frac{\partial SSE}{\epsilon_{i^*,j^*}} = 2\sum_{k=1}^{N_s}\Big[&c_{i^*,j^*,k}\{R(s_{i^*},a_{j^*})+\gamma(1-\epsilon_{i^*,j^*})Q^{\pi}(s_k,\pi(s_k))\\
&+\epsilon_{i^*,j^*}\frac{\gamma}{N_s}\underbrace{\sum_{k'=1}^{N_s} Q^{\pi}(s_{k'},\pi(s_{k'}))}_{\text{constant}\forall(s_{i^*},a_{j^*})}-Q^{\pi}(s_{i^*},a_{j^*})\}\{-\gamma Q^{\pi}(s_k,\pi(s_k))\\
&+\frac{\gamma}{N_s}\underbrace{\sum_{k'=1}^{N_s} Q^{\pi}(s_{k'},\pi(s_{k'}))\}}_{\text{constant}\forall(s^*,a^*)}\Big]=0
\end{aligned}
$$

$\frac{1}{N_s}\sum_{k'=1}^{N_s} Q^{\pi}(s_{k'},\pi(s_{k'}))$, the average value across states under policy π, is constant, so replace with v^{π}_{avg}.

$$
\begin{aligned}
0 = \sum_{k=1}^{N_s}\Big[&c_{i^*,j^*,k}\big(-\gamma Q^{\pi}(s_k,\pi(s_k))+\gamma v^{\pi}_{avg}\big)\\
&\big(R(s_{i^*},a_{j^*})+\gamma(1-\epsilon_{i^*,j^*})Q^{\pi}(s_k,\pi(s_k))+\epsilon_{i^*,j^*}\gamma v^{\pi}_{avg}-Q^{\pi}(s_{i^*},a_{j^*})\big)\Big]
\end{aligned}
$$

Multiply out the terms.

$$\begin{aligned}
0 = &-\gamma R(s_{i^*}, a_{j^*}) \sum_{k=1}^{N_s} c_{i^*,j^*,k} Q^\pi(s_k, \pi(s_k)) - \gamma^2 (1-\epsilon_{i^*,j^*}) \sum_{k=1}^{N_s} c_{i^*,j^*,k} Q^\pi(s_k, \pi(s_k))^2 \\
&- \gamma^2 v^\pi_{avg} \epsilon_{i^*,j^*} \sum_{k=1}^{N_s} c_{i^*,j^*,k} Q^\pi(s_k, \pi(s_n)) + \gamma Q^\pi(s_{i^*}, a_{j^*}) \sum_{k=1}^{N_s} c_{i^*,j^*,k} Q^\pi(s_k, \pi(s_k)) \\
&+ \gamma v^\pi_{avg} R(s_{i^*}, a_{j^*}) \sum_{k=1}^{N_s} c_{i^*,j^*,k} + \gamma^2 v^\pi_{avg} (1-\epsilon_{i^*,j^*}) \sum_{k=1}^{N_s} c_{i^*,j^*,k} Q^\pi(s_k, \pi(s_k)) \\
&+ \gamma^2 (v^\pi_{avg})^2 \epsilon_{i^*,j^*} \sum_{k=1}^{N_s} c_{i^*,j^*,k} - \gamma v^\pi_{avg} Q^\pi(s_{i^*}, a_{j^*}) \sum_{k=1}^{N_s} c_{i^*,j^*,k}
\end{aligned}$$

Write for all (s,a) in matrix form. We use the following notation for the matrix equation: $\mathcal{E}$ is an $N_s N_a \times N_s N_a$ matrix with $\epsilon_{i,j}$ across the main diagonal, C is the $N_s N_a \times N_s$ matrix of transition counts (one row for every (s,a), one column for every s'), Π is a matrix mapping Q^π to V^π for fixed policy π.

$$\begin{aligned}
\vec{0} = &-\gamma Diag(R) C \Pi^T Q^\pi - \gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}})(\mathbf{1}_{N_s N_a \times 1} - \mathcal{E}) \\
&- \gamma^2 v^\pi_{avg} Diag(C \Pi^T Q^\pi) \mathcal{E} + \gamma Diag(C \Pi^T Q^\pi) Q^\pi \\
&+ \gamma v^\pi_{avg} Diag(R) C \mathbf{1}_{N_s \times 1} + \gamma^2 k v^\pi_{avg} Diag(C \Pi^T Q^\pi)(\mathbf{1}_{N_s N_a \times 1} - \mathcal{E}) \\
&+ \gamma^2 (v^\pi_{avg})^2 Diag(C \mathbf{1}_{N_s \times 1}) \mathcal{E} - \gamma v^\pi_{avg} Diag(C \mathbf{1}_{N_s \times 1}) Q^\pi
\end{aligned}$$

Solve for $\mathcal{E}$.

$$\begin{aligned}
&\gamma Diag(R) C \Pi^T Q^\pi + \gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}}) \mathbf{1}_{N_s N_a \times 1} - \gamma Diag(C \Pi^T Q^\pi) Q^\pi \\
&- \gamma v^\pi_{avg} Diag(R) C \mathbf{1}_{N_s \times 1} - \gamma^2 v^\pi_{avg} Diag(C \Pi^T Q^\pi) \mathbf{1}_{N_s N_a \times 1} + \gamma v^\pi_{avg} Diag(C \mathbf{1}_{N_s \times 1}) Q^\pi \\
&= \gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}}) \mathcal{E} - \gamma^2 v^\pi_{avg} Diag(C \Pi^T Q^\pi) \mathcal{E} - \gamma^2 v^\pi_{avg} Diag(C \Pi^T Q^\pi) \mathcal{E} \\
&+ \gamma^2 (v^\pi_{avg})^2 Diag(C \mathbf{1}_{N_s \times 1}) \mathcal{E}
\end{aligned}$$

Pull out factor of $\mathcal{E}$

$$\begin{aligned}
&\gamma Diag(R) C \Pi^T Q^\pi + \gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}}) \mathbf{1}_{N_s N_a \times 1} - \gamma Diag(C \Pi^T Q^\pi) Q^\pi \\
&- \gamma v^\pi_{avg} Diag(R) C \mathbf{1}_{N_s \times 1} - \gamma^2 v^\pi_{avg} Diag(C \Pi^T Q^\pi) \mathbf{1}_{N_s N_a \times 1} + \gamma v^\pi_{avg} Diag(C \mathbf{1}_{N_s \times 1}) Q^\pi \\
&= \Big[\gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}}) - \gamma^2 v^\pi_{avg} Diag(C \Pi^T Q^\pi) \\
&- \gamma^2 v^\pi_{avg} Diag(C \Pi^T Q^\pi) + \gamma^2 (v^\pi_{avg})^2 Diag(C \mathbf{1}_{N_s \times 1}) \Big] \mathcal{E}
\end{aligned}$$

$$\mathcal{E} = \Big[\gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}}) - 2\gamma^2 v^\pi_{avg} Diag(C\Pi^T Q^\pi) + \gamma^2 (v^\pi_{avg})^2 Diag(C\mathbf{1}_{N_s\times 1})\Big]^{-1}$$

$$\Big[\gamma Diag(R)C\Pi^T Q^\pi + \gamma^2 Diag(C \underbrace{[\Pi^T Q^\pi]_{sq}}_{\text{elementwise square}})\mathbf{1}_{N_sN_a\times 1} - \gamma Diag(C\Pi^T Q^\pi)Q^\pi$$

$$- \gamma v^\pi_{avg} Diag(R)C\mathbf{1}_{N_s\times 1} - \gamma^2 v^\pi_{avg} Diag(C\Pi^T Q^\pi)\mathbf{1}_{N_sN_a\times 1} + \gamma v^\pi_{avg} Diag(C\mathbf{1}_{N_s\times 1})Q^\pi\Big]$$

Now we have an expression for ϵ in terms of Q^π. Algorithm 1 alternates between updating $Q(s,a)$ by linear regression and updating $\epsilon^*_{i,j}$ by the equation above.

D.3 State-Action-Specific Regularization Parameter Calculation Method 2

As an alternative method, we also calculated $\mathcal{E}$ using the assumption that the count data follows a multinomial distribution, as we did in the model-based case.

True Q: $Q^\pi(s_i,a_j) = R(s_i,a_j) + \gamma\sum_{k=1}^{N_s} T(s_j,a_j,s_k)v^\pi(s_k)$

Regularized Q:

$Q^\pi_\epsilon(s_i,a_j) = R(s_i,a_j) + \gamma(1-\epsilon_{i,j})\sum_{k=1}^{N_s} T(s_i,a_j,s_k)V^\pi(s_k) + \gamma\epsilon_{i,j}\frac{1}{N_s}\sum_{k=1}^{N_s} V^\pi(s_k)$

Bias$(\hat{Q}^\pi_\epsilon(s_i,a_j)) = \gamma\epsilon_{i,j}\Big[\sum_{k=1}^{N_s}\frac{1}{N_s}V(s_k) - \sum_{k=1}^{N_s} T(s_i,a_j,s_k)V^\pi(s_k)\Big]$

Variance$(\hat{Q}^\pi_\epsilon(s_i,a_j))$: Consider fixed V^π. Then,

$$\begin{aligned}\text{Var}(\hat{Q}^\pi_\epsilon(s_i,a_j)) &= \text{Var}\big(\gamma(1-\epsilon_{i,j})\sum_{k=1}^{N_s}\hat{T}(s_i,a_j,s_k)V^\pi(s_k)\big)\\ &= \text{Var}\big(\gamma(1-\epsilon_{i,j})\sum_{k=1}^{N_s}\frac{c_{i,j,k}}{\sum_{k'} c_{i,j,k'}}V^\pi(s_k)\\ &= \frac{\gamma^2(1-\epsilon_{i,j})^2}{(\sum_{k'=1}^{N_s} c_{i,j,k'})^2}\text{Var}\big(\sum_{k=1}^{N_s} c_{i,j,k}V^\pi(s_k)\big)\end{aligned}$$

Expand using the variance and covariance of a multinomial distribution. Let $c_{i,j} = \sum_{k=1}^{N_s} c_{i,j,k}$, the number of transition counts in the data starting at state s_i, taking action a_k, and transitioning to any state.

$$= \frac{\gamma^2(1-\epsilon_{i,j})^2}{c_{i,j}^2}\Big[\sum_{k=1}^{N_s}(V^\pi(s_k))^2 c_{i,j}T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))$$

$$-2\sum_{k=1}^{N_s}\sum_{k'\neq k} c_{i,j}V^\pi(s_k)V^\pi(s_{k'})T(s_i,a_j,s_k)T(s_i,a_j,s_{k'})\Big]$$

$$= \frac{\gamma^2(1-\epsilon_{i,j})^2}{c_{i,j}}\Big[\sum_{k=1}^{N_s}(V^\pi(s_k))^2 T(s_i,a_j,s_k)(1-T(s_i,a_j,s_k))$$

$$-2\sum_{k=1}^{N_s}\sum_{k'\neq k} V^\pi(s_k)V^\pi(s_{k'})T(s_i,a_j,s_k)T(s_i,a_j,s_{k'})\Big]$$

$T(s_i, a_j, s_k)$ is the true transition probability. We repeated our empirical experiments choosing the value of $\epsilon_{i,j}$ that minimizes the MSE, $\text{MSE}(\hat{Q}^{\pi}_{\epsilon}(s_i, a_j)) = \text{Bias}^2(\hat{Q}^{\pi}_{\epsilon}(s_i, a_j)) + \text{Var}(\hat{Q}^{\pi}_{\epsilon}(s_i, a_j))$, using the true value of T to compute an upper bound on the performance of the method. Results were similar to those using $\epsilon_{i,j}$ as calculated in Sec. D.2.

Appendix E. Model-Free Additional Empirical Results

The performance of our regularization methods reflect the fact that model-based methods generally perform better than model-free methods in low-data settings. As we increase the size of the data set used to compute the optimal policy, demonstrated in Fig. 11, the performance of the model-free methods approach that of the model-based methods. However, at the point where there is sufficient data for the model-free methods to perform well, regularization is no longer needed.

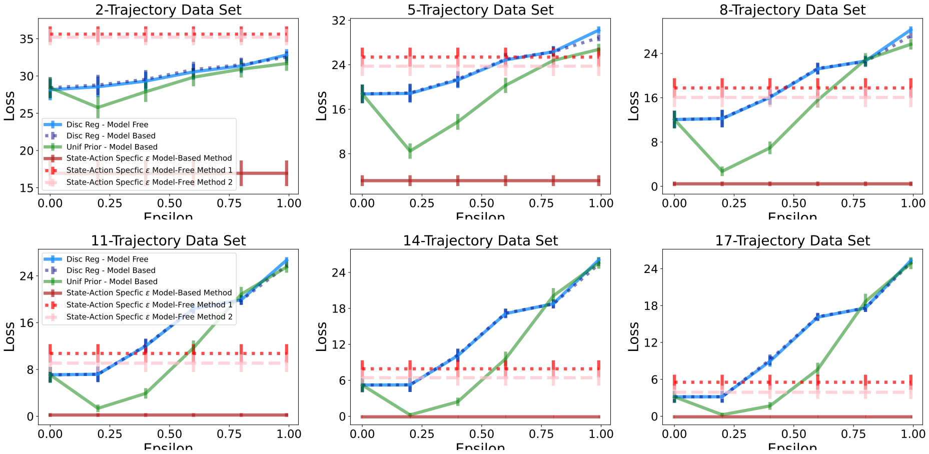
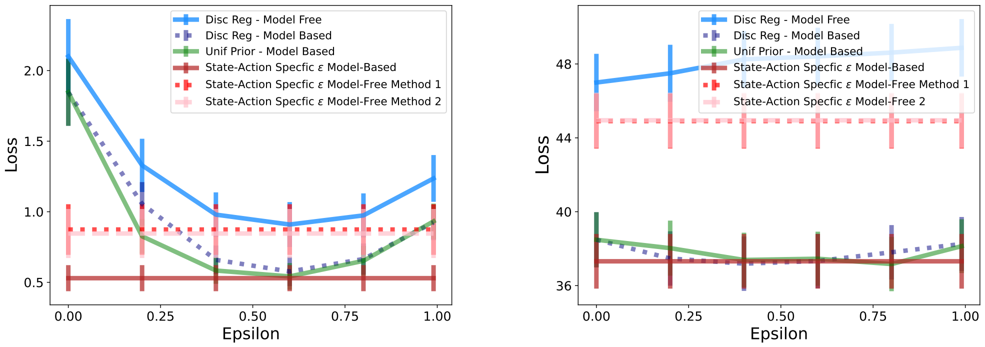


Figure 11: *River Swim Environment.* Comparison of model-free and model-based regularization methods for data sets of varying size, each with trajectories of length 20. State-action-specific methods show loss without estimation error, using true value of T or Q.

Fig. 12 compares our methods (again without estimation error), on the other two environments, showing that performance is worse for the model-free methods compared with model-based in all cases. The case of the Random Chain environment is most promising. However, even for this environment, the lower loss in comparison to global parameter methods was not replicated when using estimated values of Q to compute $\epsilon^*_{i,j}$, as demonstrated in Fig. 13.



(a) 10-State Random Chain

(b) GridWorld

Figure 12: Comparison of model-free and model-based regularization methods. State-action-specific methods show loss without estimation error, using true value of T or Q

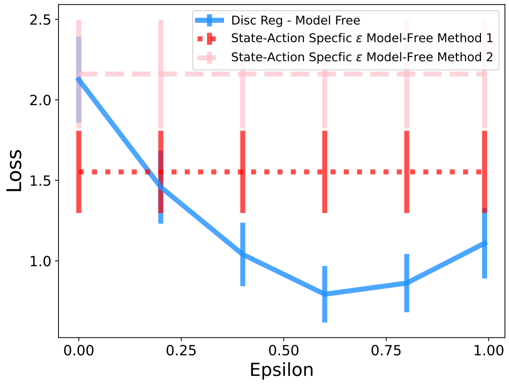


Figure 13: *Random Chain Environment.* Comparison of discount regularization and our state-action specific regularization in model-free RL.

Appendix F. Continuous State Space State-Action-Specific Regularization

We model the expected next state given current state and action using the Nadaraya-Watson kernel estimator,

$$\hat{T}_{NW}(s, a_k) = \frac{\sum_{i=1}^{D} K_h(s - s_i)s'_i}{\sum_i K_h(s - s_i)}$$

for D data tuples $\{s, a_k, s'\}_{i=1}^{D}$ for given action $a = a_k$. K_h is a kernel function of lengthscale h.

Our regularized estimate of next state is:

$$\hat{T}_\epsilon(s, a_k) = (1-\epsilon)\hat{T}_{NW}(s, a_k) + \epsilon T_{reg}$$

where T_{reg} is the mean of the chosen regularization transition distribution. We choose the value of ϵ that minimizes the MSE of $\hat{T}_\epsilon(s, a_k)$ separately for any state-action pair. To estimate the bias and variance of $\hat{T}_\epsilon(s, a_k)$, we use the estimates of bias and variance implied by the approximate kernel regression confidence bounds provided in Wasserman (2004).

Bias The approximate kernel regression confidence bounds are symmetric, implying zero bias, so the bias in $\hat{T}_\epsilon(s, a_k)$ comes from the second term only.

$$\text{Bias}(\hat{T}_\epsilon(s, a_k)) \approx \epsilon(T_{reg} - \hat{T}_{NW}(s, a_k))$$

Variance The approximate kernel regression confidence bounds use a standard error of

$$\hat{se}(s, a_k) = \hat{\sigma}\sqrt{\sum_{i=1}^{D}\Big[\frac{K_h(s-s_i)}{\sum_{j=1}^{D} K_h(s-s_j)}\Big]^2}$$

Where $\hat{\sigma}^2 = \frac{1}{2(n-1)}\sum_{d=1}^{D-1}(s'_{d+1} - s'_d)^2$ and the data for each action is ordered by the values of $\{s_d\}$.

T_{reg} is independent of the data, so the second term does not contribute to the variance. This implies the following variance of our regularized estimator:

$$\text{Variance}(\hat{T}_\epsilon(s, a_k)) \approx (1-\epsilon)^2\hat{se}^2(s, a_k)$$

MSE

$$\begin{aligned}
\text{MSE}(\hat{T}_\epsilon(s, a_k)) \approx & f(\epsilon) = \epsilon^2(T_{reg} - \hat{T}_{NW}(s, a_k))^2 + (1-\epsilon)^2\hat{se}^2(s, a_k) \\
& f'(\epsilon) = 2\epsilon(T_{reg} - \hat{T}_{NW}(s, a_k))^2 - 2(1-\epsilon)\hat{se}^2(s, a_k) = 0 \\
& 0 = \epsilon 2(T_{reg} - \hat{T}_{NW}(s, a_k))^2 - 2\hat{se}^2(s, a_k) + 2\epsilon\hat{se}^2(s, a_k) \\
& \hat{se}^2(s, a_k) = \epsilon[(T_{reg} - \hat{T}_{NW}(s, a_k))^2 + \hat{se}^2(s, a_k)] \\
& \epsilon = \frac{\hat{se}^2(s, a_k)}{(T_{reg} - \hat{T}_{NW}(s, a_k))^2 + \hat{se}^2(s, a_k)}
\end{aligned}$$

Appendix G. Continuous State Space Empirical Example Details

The empirical example used to demonstrate the continuous state-action specific regularization algorithm in Sec. 7.2 is a continuous-state version of the River Swim environment described in Sec. 6.1.

State Space We consider a one-dimensional state space, $s \in [0, 1]$.

Discount Factor We evaluate all policies using a true discount factor of $\gamma = 0.99$.

Reward Function We define the reward function $R(s) = (s - 0.5)^2 + s(s - .5)^3$. As illustrated in Fig. 14, this results in a smaller positive reward near the lower end of the state space and a large postive reward at the upper end of the state space.

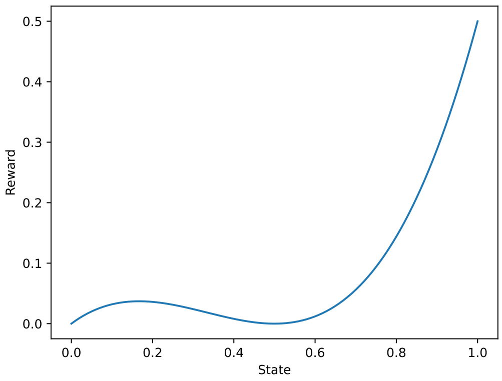


Figure 14: Continuous river swim environment reward function.

Actions There are two discrete actions. Action 0 moves the agent "upstream" towards the larger reward with greater stochasticity. Action 1 moves the agent "downstream" towards the smaller reward with less stochasticity. The transition distributions for each action are described below.

Transition Function

$$T_{A0}(s, \text{noise\_level}) = \begin{cases} s' \sim \text{Unif}[0,1] \text{ with probability } 0.50 \\ s' = s + 0.20 + \text{noise\_level} \times \text{Unif}[0,1] \text{ with probability } 0.45 \\ s' = s - 0.20 + \text{noise\_level} \times \text{Unif}[0,1] \text{ with probability } 0.05 \end{cases}$$

$$T_{A1}(s, \text{noise\_level}) = \begin{cases} s' \sim \text{Unif}[0,1] \text{ with probability } 0.25 \\ s' = s - 0.20 + \text{noise\_level} \times \text{Unif}[0,1] \text{ with probability } 0.75 \end{cases}$$

G.1 Extensions to Online RL

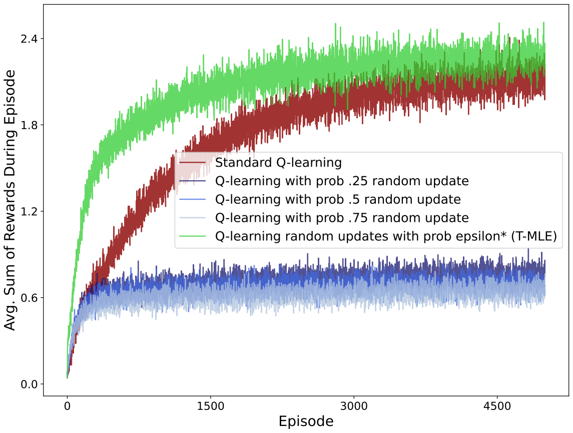
The examples in the paper apply our ideas to offline RL, however it is possible to apply the same concepts to online RL. For example, we can modify the Q-learning algorithm to incorporate a weighted average update rule at each step, as demonstrated in Algorithm 3 and associated results in Fig. 15.

Appendix H. Code Repository

Python code to reproduce the results in this paper is available at at: `https://github.com/dtak/rethinking_discount_reg_public`.

Algorithm 3 Q-learning with State-Action-Specific Regularization

```
Parameters: step size $\alpha \in (0, 1]$, regularization matrix $T_{reg}(s, a)$
Initialize $Q(s, a) = 0 \forall (s, a)$
for $e = 1$ to [number of episodes] do
    Choose initial state $s$ randomly.
    while step_counter < [steps per episode] do
        Choose action $a$ from policy based on $Q(s, a)$ (e.g. epsilon-greedy based on current Q)
        Calculate $\epsilon^*$ from Eq. 7
        Draw $x \sim Bernoulli(\epsilon^*)$
        if $x = 1$ then
            Draw simulated next step $s'_{sim}$ from $T_{reg}(s, a)$
            Update Q-function using $s'_{sim}$:
            $Q(s, a) \leftarrow Q(s, a) + \alpha[r(s, a) + \gamma \max_a Q(s'_{sim}, A) - Q(s, a)]$
        else if $x = 0$ then
            Agent takes action $a$, observes next state $s'$
            $Q(s, a) \leftarrow Q(s, a) + \alpha[r(s, a) + \gamma \max_a Q(s', A) - Q(s, a)]$
            step_counter += 1
            $s \leftarrow s'$
        end if
    end while
end for
```



(a) River Swim

Figure 15: *River Swim Environment.* Comparison of state-action-specific regularization to standard Q-learning in our three tabular environments. We also include as a baseline our algorithm with ϵ^* replaced by a constant probability.

References

Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, pages 10–4, 2019.

Ron Amit, Ron Meir, and Kamil Ciosek. Discount factor as a regularizer in reinforcement learning. In *International conference on machine learning*, pages 269–278. PMLR, 2020.

John Asmuth, Lihong Li, Michael L Littman, Ali Nouri, and David Wingate. A bayesian sampling approach to exploration in reinforcement learning. *arXiv preprint arXiv:1205.2664*, 2012.

Raghav Awasthi, Keerat Kaur Guliani, Saif Ahmad Khan, Aniket Vashishtha, Mehrab Singh Gill, Arshita Bhatt, Aditya Nagori, Aniket Gupta, Ponnurangam Kumaraguru, and Tavpritesh Sethi. Vacsim: Learning effective strategies for covid-19 vaccine distribution using reinforcement learning. *Intelligence-Based Medicine*, page 100060, 2022.

William Cai, Josh Grossman, Zhiyuan Jerry Lin, Hao Sheng, Johnny Tian-Zheng Wei, Joseph Jay Williams, and Sharad Goel. Bandit algorithms to personalize educational chatbots. *Machine Learning*, 110(9):2389–2418, 2021.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 1282–1289. PMLR, 2019.

Michael O'Gordon Duff. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes.* University of Massachusetts Amherst, 2002.

Audrey Durand, Charis Achilleos, Demetris Iacovides, Katerina Strati, Georgios D Mitsis, and Joelle Pineau. Contextual bandits for adapting treatment in a mouse model of de novo carcinogenesis. In *Machine learning for healthcare conference*, pages 67–82. PMLR, 2018.

Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6, 2005.

Benjamin Eysenbach, Matthieu Geist, Sergey Levine, and Ruslan Salakhutdinov. A connection between one-step rl and critic regularization in reinforcement learning. In *International Conference on Machine Learning*, pages 9485–9507. PMLR, 2023.

Jesse Farebrother, Marlos C Machado, and Michael Bowling. Generalization and regularization in dqn. *arXiv preprint arXiv:1810.00123*, 2018.

Mohammad Ghavamzadeh, Alessandro Lazaric, Rémi Munos, and Matt Hoffman. Finite-sample analysis of lasso-td. In *International Conference on Machine Learning*, 2011.

Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, Aviv Tamar, et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6): 359–483, 2015.

GC Goodwin and KS Sin. Adaptive filtering prediction and control,(book) prentice-hall. *Englewood Cliffs*, 6(7):45, 1984.

Omer Gottesman, Joseph Futoma, Yao Liu, Sonali Parbhoo, Leo Celi, Emma Brunskill, and Finale Doshi-Velez. Interpretable off-policy evaluation in reinforcement learning by highlighting influential transitions. In *International Conference on Machine Learning*, pages 3658–3667. PMLR, 2020.

Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1181–1189. Citeseer, 2015.

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

J Zico Kolter and Andrew Y Ng. Near-bayesian exploration in polynomial time. In *Proceedings of the 26th annual international conference on machine learning*, pages 513–520, 2009a.

J Zico Kolter and Andrew Y Ng. Regularization and feature selection in least-squares temporal difference learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 521–528, 2009b.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. In *AI&M*, 2006.

Peng Liao, Kristjan Greenewald, Predrag Klasnja, and Susan Murphy. Personalized heartsteps: A reinforcement learning algorithm for optimizing physical activity. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–22, 2020.

Bo Liu, Sridhar Mahadevan, and Ji Liu. Regularized off-policy td-learning. *Advances in Neural Information Processing Systems*, 25, 2012.

Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. Regularization matters in policy optimization–an empirical study on continuous control. *arXiv preprint arXiv:1910.09191*, 2019.

Kevin P Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Elizbar A Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9 (1):141–142, 1964.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287, 1999.

Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and q-learning. *arXiv preprint arXiv:1611.01626*, 2016.

Brendan O'Donoghue, Ian Osband, and Catalin Ionescu. Making sense of reinforcement learning and probabilistic inference. *arXiv preprint arXiv:2001.00805*, 2020.

Sang Ho Oh, Jongyoul Park, Su Jin Lee, Seungyeon Kang, and Jeonghoon Mo. Reinforcement learning-based expanded personalized diabetes treatment recommendation using south korean electronic health records. *Expert Systems with Applications*, 206:117932, 2022.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper/2013/file/6a5889bb0190d0211a991f47bb19a777-Paper.pdf`.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Silviu Pitis. Rethinking the discount factor in reinforcement learning: A decision theoretic approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7949–7956, 2019.

Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

Pascal Poupart, Nikos Vlassis, Jesse Hoey, and Kevin Regan. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pages 697–704, 2006.

Yi Qi, Qingyun Wu, Hongning Wang, Jie Tang, and Maosong Sun. Bandit learning with implicit feedback. *Advances in Neural Information Processing Systems*, 31, 2018.

Sarah Rathnam, Sonali Parbhoo, Weiwei Pan, Susan Murphy, and Finale Doshi-Velez. The unintended consequences of discount regularization: Improving regularization in certainty equivalence reinforcement learning. In *International Conference on Machine Learning*, pages 28746–28767. PMLR, 2023.

Benjamin Ribba, Gentian Kaloshi, Mathieu Peyre, Damien Ricard, Vincent Calvez, Michel Tod, Branka Čajavec-Bernard, Ahmed Idbaih, Dimitri Psimaras, Linda Dainese, et al. A tumor growth inhibition model for low-grade glioma treated with chemotherapy or radiotherapya tumor growth inhibition model for low-grade glioma. *Clinical Cancer Research*, 18(18):5071–5080, 2012.

Stephane Ross, Brahim Chaib-draa, and Joelle Pineau. Bayes-adaptive pomdps. *Advances in neural information processing systems*, 20, 2007.

Stéphane Ross, Joelle Pineau, Brahim Chaib-draa, and Pierre Kreitmann. A bayesian approach for learning and planning in partially observable markov decision processes. *Journal of Machine Learning Research*, 12(5), 2011.

Jonathan Sorg, Satinder Singh, and Richard L Lewis. Variance-based rewards for approximate bayesian reinforcement learning. *arXiv preprint arXiv:1203.3518*, 2012.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, page 113. MIT press, 2018.

Csaba Szepesvári and Rémi Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd international conference on Machine learning*, pages 880–887, 2005.

Anna L Trella, Kelly W Zhang, Inbal Nahum-Shani, Vivek Shetty, Finale Doshi-Velez, and Susan A Murphy. Designing reinforcement learning algorithms for digital interventions: pre-implementation guidelines. *Algorithms*, 15(8):255, 2022.

Nikos Vlassis, Mohammad Ghavamzadeh, Shie Mannor, and Pascal Poupart. Bayesian reinforcement learning. *Reinforcement learning*, pages 359–386, 2012.

Larry Wasserman. *All of statistics: a concise course in statistical inference*, volume 26. Springer, 2004.

Geoffrey S Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 359–372, 1964.

Qingda Wei and Xianping Guo. Markov decision processes with state-dependent discount factors and unbounded rewards/costs. *Operations Research Letters*, 39(5):369–374, 2011.

Martha White. Unifying task specification in reinforcement learning. In *International Conference on Machine Learning*, pages 3742–3750. PMLR, 2017.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.

Naoto Yoshida, Eiji Uchibe, and Kenji Doya. Reinforcement learning with state-dependent discount factor. In *2013 IEEE third joint international conference on development and learning and epigenetic robotics (ICDL)*, pages 1–6. IEEE, 2013.

Yang Yu. Towards sample efficient reinforcement learning. In *IJCAI*, pages 5739–5743, 2018.

Andrea Zanette and Emma Brunskill. Problem dependent reinforcement learning bounds which can identify bandit structure in mdps. In *International Conference on Machine Learning*, pages 5747–5755. PMLR, 2018.