

A. Near-Optimal SVP With Additive Near-Optimality

We can quantify the near-optimality of any given SVP π by using a version of the performance difference lemma (Kakade & Langford, 2002).

Theorem 3. For any SVP π , if for every state $s \in \mathcal{S}$:

$$\max_{a \in \pi(s)} (Q^*(s, a^*) - Q^*(s, a)) \leq \epsilon ,$$

then

$$V^*(s) - V^\pi(s) \leq \frac{\epsilon}{1 - \gamma} .$$

Proof. Note that $V^*(s) \leq V^\pi(s)$ for any SVP π (equality holds when the SVP π corresponds to an optimal policy). Denoting $a^* = \pi^*(s)$, and $\bar{a} = \arg \min_{a \in \pi(s)} Q^\pi(s, a)$. We can evaluate the difference between V^* and V^π for a particular state s :

$$\begin{aligned} V^*(s) - V^\pi(s) &= [r(s, a^*) + \gamma \mathbb{E}_{s'|s, a^*} V^*(s')] - [r(s, \bar{a}) + \gamma \mathbb{E}_{s'|s, \bar{a}} V^\pi(s')] \\ &= [r(s, a^*) + \gamma \mathbb{E}_{s'|s, a^*} V^*(s')] - [r(s, \bar{a}) + \gamma \mathbb{E}_{s'|s, \bar{a}} V^*(s')] \\ &\quad + [r(s, \bar{a}) + \gamma \mathbb{E}_{s'|s, \bar{a}} V^*(s')] - [r(s, \bar{a}) + \gamma \mathbb{E}_{s'|s, \bar{a}} V^\pi(s')] \\ &\quad \text{(adding and subtracting the expressions in blue)} \\ &= [Q^*(s, a^*) - Q^*(s, \bar{a})] + \gamma [\mathbb{E}_{s'|s, \bar{a}} (V^*(s') - V^\pi(s'))] . \end{aligned}$$

Suppose we are guaranteed that for every state s , we have the following bound on the action-value gap of actions in $\pi(s)$:

$$\max_{a \in \pi(s)} (Q^*(s, a^*) - Q^*(s, a)) \leq \epsilon ,$$

or equivalently,

$$\forall a \in \pi(s) : Q^*(s, a^*) - Q^*(s, a) \leq \epsilon ,$$

we can further simplify:

$$V^*(s) - V^\pi(s) \leq \epsilon + \gamma [\mathbb{E}_{s'|s, \bar{a}} (V^*(s') - V^\pi(s'))] .$$

Unrolling the recursive expression:

$$\begin{aligned} V^*(s) - V^\pi(s) &\leq \epsilon + \gamma \epsilon + \gamma^2 [\mathbb{E}_{s''|s, \bar{a}, s', \bar{a}'} (V^*(s'') - V^\pi(s''))] \\ &\leq \epsilon + \gamma \epsilon + \gamma^2 \epsilon + \dots \\ &= \frac{\epsilon}{1 - \gamma} . \end{aligned} \quad \square$$

If we want the worst-case values of π to be within a margin defined as some fraction ζ of the maximum magnitude optimal value, e.g.,

$$\max_s (V^*(s) - V^\pi(s)) \leq \zeta \|V^*\|_\infty ,$$

then we can set

$$\frac{\epsilon}{1 - \gamma} = \zeta \|V^*\|_\infty ,$$

which implies that the action-value gap should be upper bounded by $\epsilon = (1 - \gamma)\zeta \|V^*\|_\infty$. In practice, once we learn Q^* and V^* , we can construct the SVP as

$$\pi(s) = \{a : Q^*(s, a) \geq V^*(s) - \epsilon\} .$$

B. Learning SVPs via an Exponential Action Space – And Why It Does Not Work

Alternatively, one might reformulate the task of learning SVPs by considering an exponentially large action space, $\tilde{\mathcal{A}} = 2^{\mathcal{A}} \setminus \{\emptyset\}$. By applying standard approaches to an MDP with this new action space, one can learn a policy π that maps each state to an element of $\tilde{\mathcal{A}}$, such that $\pi(s) = \tilde{a}$. Under this formulation, Q-values are defined over $\mathcal{S} \times \tilde{\mathcal{A}}$, which we denote $Q^\pi(s, \tilde{a})$. Consider the worst-case Q-values defined analogously to [Definition 2](#):

$$Q^\pi(s, \tilde{a}) = \min_{a \in \tilde{a}} Q^\pi(s, a).$$

Then, for any $\tilde{a} \in \tilde{\mathcal{A}}$, we have

$$Q^\pi(s, \tilde{a}) = \min_{a \in \tilde{a}} Q^\pi(s, a) \leq \max_{a \in \tilde{a}} Q^\pi(s, a),$$

and since $Q^\pi(s, a) = Q^\pi(s, \{a\})$, there exists an \tilde{a}^* such that

$$Q^\pi(s, \tilde{a}) \leq Q^\pi(s, \tilde{a}^*) \text{ where } \tilde{a}^* = \left\{ \arg \max_{a \in \tilde{\mathcal{A}}} Q^\pi(s, a) \right\}.$$

Intuitively, selecting the best action in \tilde{a} is always *no worse* than selecting the worst action in \tilde{a} . This suggests that for any non-singleton set action \tilde{a} , we can always find a singleton set action \tilde{a}^* that is better. Thus, this formulation results in trivial SVPs and does not discover near-equivalent actions. To yield meaningful solutions, one would require additional constraints.

C. Example: Non-Existence of Near-Greedy SVP Fixed-Point

Recall the near-greedy fixed-point equation:

$$\pi(s) = \{a : Q^\pi(s, a) \geq (1 - \zeta)V^*(s)\} \text{ where } Q^\pi(s, a) = r(s, a) + \gamma \mathbb{E}_{s'|s, a} \left[\min_{a' \in \pi(s')} \{Q^\pi(s', a')\} \right]$$

Consider the MDP in [Figure 8](#) with two non-terminal states $\{s_1, s_2\}$ and two actions $\{L, R\}$. Let $\gamma = 0.9$, $\zeta = 0.2$. Here, $V^* = [0.9, 1]$. There are two candidate SVPs, both of which fail to satisfy the near-greedy fixed-point equation.

- Suppose $\pi(s_1) = \{R\}$, $\pi(s_2) = \{R\}$. Then $Q^\pi(s_2, L) = 0.81 > (1 - \zeta)V^*(s_2)$, meaning that L is a near-optimal action at s_2 but not included in $\pi(s_2)$.
- Suppose $\pi(s_1) = \{R\}$, $\pi(s_2) = \{L, R\}$. Then the worst-case $Q^\pi(s_2, L) = 0$ because the agent falls into a cycle in the worst case, and thus L is not a near-optimal action but is included in $\pi(s_2)$.

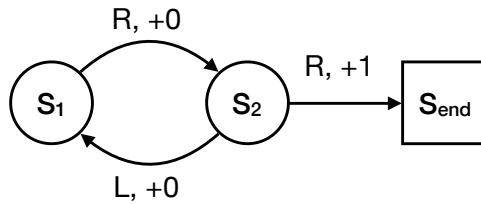


Figure 8. A three-state MDP with no near-greedy fixed-point solution when $\gamma = 0.9$ and $\zeta = 0.2$.

D. More on the Conservative Heuristic

Theorem 4. *The conservative ζ -optimal SVP exists and is unique for any MDP with non-negative rewards.*

Proof. In the conservative heuristic, there is no recursive relationship between policy π and its value function V^π or Q^π . The policy construction depends on the lower-bound action-value function \check{Q}^* , which computes an expectation over V^* and immediate rewards r , and is thus unique, and so is π .

To show that π is a valid SVP, we will show that the optimal action at every state is always included in $\pi(s)$ such that $\forall s \in \mathcal{S}, \pi(s) \neq \emptyset$. Consider the optimal action at state s , $a^* = \arg \max_{a \in \mathcal{A}} Q^*(s, a)$ where $V^*(s) = Q^*(s, a^*)$, we have:

$$\begin{aligned}
 \check{Q}_\zeta^*(s, a^*) &= r(s, a^*) + \gamma \mathbb{E}_{s'|s, a^*} (1 - \zeta) V^*(s') \\
 &= \zeta r(s, a^*) + (1 - \zeta) [r(s, a^*) + \gamma \mathbb{E}_{s'|s, a^*} V^*(s')] \\
 &= \zeta r(s, a^*) + (1 - \zeta) Q^*(s, a^*) \\
 &\geq (1 - \zeta) Q^*(s, a^*) \\
 &= (1 - \zeta) V^*(s).
 \end{aligned}$$

□

Since the conservative heuristic calculates an expectation over V^* and $r(s, a)$ and does not involve any recursive relationship, after learning Q^* (and thus V^*), we can apply a standard stochastic approximation algorithm with provable convergence guarantees (Robbins & Monro, 1951). While the conservative heuristic has good theoretical properties, in Section 5.1 we observe that it does not discover as many near-optimal actions compared to near-greedy (due to it being conservative).

E. Convergence Analysis for the Near-Greedy TD Algorithm (Algorithm 1)

E.1. Contraction

For the case of a general MDP (possibly non-DAG), we refer to the convergence proofs of TD methods such as Q-learning and expected SARSA, which have been extensively studied in the tabular setting for problems with discrete state and action spaces (Watkins & Dayan, 1992; Melo, 2001; Van Seijen et al., 2009). For Q-learning, given bounded rewards, Q converges to the optimal value function Q^* , i.e., $Q(s, a) \simeq Q^*(s, a)$ for all $s \in \mathcal{S}$, $a \in \mathcal{A}$ with probability 1, under regular conditions for stochastic approximation: each (s, a) is updated infinitely many times, $\sum_t \alpha_t = \infty$, and $\sum_t \alpha_t^2 < \infty$. One of the key steps in the proof involves showing that the update operator H is a contraction with respect to sup-norm (Melo, 2001):

$$\begin{aligned}
 &\text{Update operator:} \\
 (HQ)(s, a) &= r(s, a) + \gamma \mathbb{E}_{s'|s, a} \max_{a' \in \mathcal{A}} Q(s', a') \\
 &\text{based on the Bellman optimality equation, and} \\
 \|HQ_1 - HQ_2\|_\infty &\leq \gamma \|Q_1 - Q_2\|_\infty.
 \end{aligned}$$

Since the proposed algorithms have the same structure as TD learning, ideally we would have the same convergence guarantees. Consider the following update operator for the near-optimal TD algorithm:

$$\begin{aligned}
 (HQ)(s, a) &= r(s, a) + \gamma \mathbb{E}_{s'|s, a} \min_{a' \in \pi(s')} Q(s', a') \\
 &\text{where } \pi(s') = \{a' : Q^\pi(s', a') \geq (1 - \zeta) V^*(s')\}.
 \end{aligned}$$

In an attempt to show that the update operator is a contraction, we can manipulate $\|HQ_1 - HQ_2\|_\infty$ in a similar way:

$$\begin{aligned}
 \|HQ_1 - HQ_2\|_\infty &= \max_{s, a} \left| \left(r(s, a) + \gamma \mathbb{E}_{s'|s, a} \min_{a'_1 \in \pi_1(s')} Q_1(s', a'_1) \right) - \left(r(s, a) + \gamma \mathbb{E}_{s'|s, a} \min_{a'_2 \in \pi_2(s')} Q_2(s', a'_2) \right) \right| \\
 &= \max_{s, a} \left| \gamma \mathbb{E}_{s'|s, a} \left[\min_{a'_1 \in \pi_1(s')} Q_1(s', a'_1) - \min_{a'_2 \in \pi_2(s')} Q_2(s', a'_2) \right] \right| \\
 &\leq \max_{s, a} \gamma \mathbb{E}_{s'|s, a} \left| \min_{a'_1 \in \pi_1(s')} Q_1(s', a'_1) - \min_{a'_2 \in \pi_2(s')} Q_2(s', a'_2) \right| \\
 &\leq \gamma \max_{s'} \left| \min_{a'_1 \in \pi_1(s')} Q_1(s', a'_1) - \min_{a'_2 \in \pi_2(s')} Q_2(s', a'_2) \right| \\
 &\leq \gamma \max_{s'} |V^*(s') - (1 - \zeta) V^*(s')| \\
 &= \gamma \zeta \max_{s'} V^*(s') = \gamma \zeta \|V^*\|_\infty.
 \end{aligned}$$

With this loose upper bound, the update operator is not necessarily a contraction, suggesting that the algorithm might not converge for a general MDP.

E.2. Convergence Proof for DAG MDPs

We first state a key result from martingale theory that we will use:

Theorem 5 (Martingale Convergence Theorem (Williams, 1991)). *Consider $\{M_n\}_{n \in \mathbb{N}}$ as martingale in \mathbb{R}^d with*

$$\sum_{n \geq 0} \mathbb{E} [\|M_{n+1} - M_n\|^2 | \mathcal{F}_n] < \infty$$

then there exists a random variable $M_\infty \in \mathbb{R}$ such that $\|M_\infty\| < \infty$ almost surely and $M_n \rightarrow_{n \rightarrow \infty} M_\infty$ almost surely.

Using this standard result, we can show the following convergence result:

Theorem 2 (restated). *The near-greedy TD algorithm (Algorithm 1) converges to the unique solution if the MDP is a DAG with non-negative rewards, under the same conditions for regular TD learning: rewards have bounded variance, each (s, a) is updated infinitely many times, $\sum_t \alpha_t = \infty$, and $\sum_t \alpha_t^2 < \infty$ for each (s, a) (Watkins & Dayan, 1992; Melo, 2001).*

Proof. Given the DAG MDP, we use H to denote the maximum number of steps (depth of the topological sort tree) and (s_h, a_h) to denote a state-action pair for a particular state s_h at step h . We use $Q_t(s_h, a_h)$ to denote the Q-value estimate after episode t in Algorithm 1. In addition, we overload the notation $Q(h)$ to refer to the vector containing Q-values of all state-action pairs $\mathcal{S}(h) \times \mathcal{A}(h)$ at step h .

From Theorem 1 we know that for a DAG MDP, the equation $\pi(s) = \{a : Q^\pi(s, a) \geq (1 - \zeta)V^*(s)\}$ has a unique fixed point solution, which we denote π^ζ and its worst-case value function as Q^ζ . Furthermore, we define the following:

$$\begin{aligned} \bar{a}(s_h) &= \arg \min_{a \in \pi^\zeta(s_h)} Q^\zeta(s_h, a) \\ \underline{a}(s_h) &= \arg \max_{a \notin \pi^\zeta(s_h)} Q^\zeta(s_h, a) \end{aligned}$$

Note that $Q^\zeta(s, \bar{a}(s)) \geq (1 - \zeta)V^*(s) \geq Q^\zeta(s, \underline{a}(s))$. Intuitively, \bar{a} gives us the worst-case action whose value will be used in the update / backup, whereas \underline{a} is the best action outside the near-optimal action set for the given ζ .

We will prove the convergence of the near-greedy TD algorithm for DAG MDPs via backward induction over the episode steps $H, H - 1, \dots, 1$.

Base step. For every terminal state s_H , the estimates are correct by initialization as $Q^\zeta(s, a) = 0$ and $\pi^\zeta(s) = \mathcal{A}$ trivially. Therefore, for all (s_H, a_H) and $\epsilon \geq 0$, there exists $t_\epsilon \geq 0$, such that, for all $t \geq t_\epsilon$, $\|Q_t(H) - Q^\zeta(H)\|_\infty \leq \epsilon$ where $Q(H)$ is the vector containing Q-values of state-action pairs at step H .

Inductive step. Assume that Q_t for all state-action pairs in levels $\{h + 1, \dots, H\}$ converge to the true Q^ζ almost surely. In other words, other than sequences of measure 0, under all possible updates, we have $Q_t(s_j, a_j) \rightarrow Q^\zeta(s_j, a_j)$ for all $j \geq h + 1$. This guarantees that for all (s_{h+1}, a_{h+1}) , for every $\epsilon > 0$, there exists $t_\epsilon > 0$ such that, for all $t \geq t_\epsilon$, $\|Q_t(h + 1) - Q^\zeta(h + 1)\|_\infty \leq \epsilon$. For notational convenience in the inductive step, we use (s, a) to denote state-action pairs at step h and (s', a') to denote state-action pairs at step $h + 1$.

Let $\Delta_1(s') = Q^\zeta(s', \bar{a}(s')) - Q^\zeta(s', \underline{a}(s'))$ and $\Delta_2(s') = \max_{a \in \pi^\zeta(s')} Q^\zeta(s', a) - Q^\zeta(s', \bar{a}(s'))$. Note that, if we pick $\epsilon < \frac{1}{2} \min_{s'} (\Delta_1(s'), \Delta_2(s'))$, then convergence implies that, for each state s' , after some episode t_0 , a constant action $\bar{a}(s')$ is used in the near-greedy update of Q-values at step h .

Consider the sequence of Q-values $\{Q_t(h)\}_{t \in \mathbb{N}}$. Let \mathcal{F}_{th} denote the history of the algorithm till step h of episode t . In our proof, we consider the updates made to $Q(h)$ after t_0 with $Q_{t_0}(h)$ as its initialization for our analysis. This reduces the proof structure to a simple stochastic approximation based argument where the constant near-greedy action is used while bootstrapping for any state s' . At any such episode $t > t_0$, the algorithm makes an update of the following form to $Q_t(h)$:

$$Q_{t+1}(s, a) = \begin{cases} Q_t(s, a) & \text{if } (s, a)_{th} \neq (s, a) \\ (1 - \alpha_{th})Q_t(s, a) + \alpha_{th}[r_{th} + \gamma Q_t(s', \bar{a}(s'))] & \text{if } (s, a)_{th} = (s, a) \end{cases}$$

We can rewrite the bootstrapping update as:

$$Q_{t+1}(s, a) = \underbrace{(1 - \alpha_{th})Q_t(s, a) + \alpha_{th}\mathbb{E}_{r, s'}[r + \gamma Q_t(s', \bar{a}(s'))]}_{\text{Bellman update}} + \underbrace{\alpha_{th}w_{th}}_{\text{noise term}} \quad (3)$$

where $w_{th} = [r_{th} + \gamma Q_t(s'_t, \bar{a}(s'_t))] - \mathbb{E}_{r,s'}[r + \gamma Q_t(s', \bar{a}(s'))]$. We now analyze these two components of the update separately.

Bellman update. First note that $\mathbb{E}_{r,s'}[(r + \gamma Q_t(s', \bar{a}(s')))]^2 < \infty$ by using the assumption $\mathbb{E}[r^2] < \infty$ and the inductive assumption on Q_t . In the near-greedy TD algorithm, for each (s, a) , the updates are made using step size α_t such that $\sum_t \alpha_t = \infty$, and $\sum_t \alpha_t^2 < \infty$. Using $\bar{Q}(s, a)$ to denote the noise-free update term in Eqn. (3), for the Bellman update sequence, we have:

$$\begin{aligned} \bar{Q}_{t+1}(s, a) - Q^\zeta(s, a) &= (1 - \alpha_{th})(Q_t(s, a) - Q^\zeta(s, a)) + \alpha_{th}\gamma\mathbb{E}_{s'}[Q_t(s', \bar{a}(s')) - Q^\zeta(s', \bar{a}(s'))] \\ &\leq (1 - \alpha_{th})(Q_t(s, a) - Q^\zeta(s, a)) + \alpha_{th}\gamma\epsilon \end{aligned}$$

where the last step follows from the inductive assumption. Using the standard results from stochastic approximation (Robbins & Monro, 1951), we can conclude that the deterministic error $\prod_{t>t_0}(1 - \alpha_{th})^2(Q_{t_0}(s, a) - Q^\zeta(s, a))^2$ converges to 0 implying $\limsup_{t \rightarrow \infty} (\bar{Q}_t(s, a) - Q^\zeta(s, a))^2 \leq C\epsilon$ for some constant C . As the chosen ϵ is arbitrary, by the sandwich theorem for limits, the error incurred via the Bellman update sequence converges to 0 almost surely.

Noise term. We will now argue that the noise sequence $\sum_{t>t_0} \alpha_{th}w_{th}$ also converges to 0. Note that, $Z_t = \sum_{t>t_0} \alpha_{th}w_{th} \in \mathbb{R}^{\mathcal{S}(h) \times \mathcal{A}(h)}$ is a martingale sequence as $\mathbb{E}[w_{th}(s, a)|\mathcal{F}_{th}] = 0$. Further, again by the bounded variance assumption over rewards and the inductive assumption over $Q(h+1)$, we have

$$\sum_{t>t_0} \mathbb{E}[\|Z_{t+1} - Z_t\|^2|\mathcal{F}_{th}] = \sum_{t>t_0} \alpha_{th}^2 \mathbb{E}[\|w_{th}\|^2|\mathcal{F}_{th}] \leq c \cdot \sum_{t>t_0} \alpha_{th}^2 \leq \infty$$

Now using Theorem 5 and the definition $Z_{t_0} = 0$, we can conclude that the martingale converges to 0 almost surely.

We know that for two sequences of random variables X_n and Y_n , if $X_n \rightarrow X$ and $Y_n \rightarrow Y$ almost surely, then $X_n + Y_n \rightarrow X + Y$ almost surely. Combining the two parts, we get $\|Q(h) - Q^\zeta(h)\|_\infty \rightarrow 0$ almost surely. This completes the inductive step.

By induction, this proves the desired convergence result. \square

F. Comparisons to the Mixed-Integer Programming (MIP) Baseline

Fard & Pineau (2011) proposed a mixed-integer programming formulation for solving the maximal-size SVP in a finite-horizon tabular planning problem. The optimization problem jointly solves for the worst-case values V and a binary representation of SVP π , where $\Pi(s, a) = \mathbb{1}[a \in \pi(s)]$ is 1 if a is an element of $\pi(s)$, and 0 otherwise. There are a total of $|\mathcal{S}|(|\mathcal{A}| + 1)$ decision variables and $|\mathcal{S}|(|\mathcal{A}| + 2)$ constraints. The formulation is reproduced below; see Fard & Pineau (2011) for more details.

$$\begin{aligned} \max_{V, \Pi} [\mu^T V + (V_{\max} - V_{\min})e_s^T \pi e_a] \text{ subject to} \\ V(s) &\geq (1 - \zeta)V^*(s) && \forall s \in \mathcal{S} \\ \sum_{a \in \mathcal{A}} \Pi(s, a) &> 0 && \forall s \in \mathcal{S} \\ V(s) &\leq r(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s'|s, a)V(s') + V_{\max}(1 - \Pi(s, a)) && \forall s \in \mathcal{S}, \forall a \in \mathcal{A} \end{aligned}$$

Since the MIP approach requires knowledge of the MDP model, we implemented a dynamic programming based approach with the near-greedy heuristic, namely near-greedy value iteration (VI). We applied these two algorithms on simple environments where the MIP solution is tractable. On Chain-5 with $\gamma = 0.9$, where the underlying MDP is a DAG (Figure 9), near-greedy VI converged for all values of ζ . The SVPs learned by both approaches satisfy near-optimality with respect to the given ζ , as shown by the worst-case near-optimality percentages. For $\zeta \geq 0.1$, the SVP included all actions at every state. Even though near-greedy VI is not explicitly maximizing the policy size (unlike the MIP approach, which includes policy size as part of its objective function), for many of the cases it still finds an SVP solution with maximal size, or close to the maximal-size solution as found by MIP (when $\zeta = 0.03$ and 0.04 on this problem). On a non-DAG environment, CyclicChain-5 with $\gamma = 0.9$ (Figure 10), near-greedy VI did not converge for $0.2 \leq \zeta < 1$ (when a near-optimal SVP should only include the two ‘right’ actions but no ‘left’ actions). This is consistent with what we observed in Figure 4a. On this problem, when near-greedy VI does converge ($\zeta \leq 0.1$, which is a suitable range of values if one aims to learn *close-to-optimal* behavior), it consistently finds the same maximal-size SVP as MIP. Compared to a model-based approach based on exhaustive search, our proposed near-greedy heuristic identifies SVP solutions that achieve good worst-case near-optimality and similar average policy sizes, despite the fact that we do not explicitly optimize for the size of the SVP.


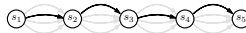




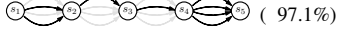





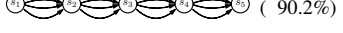





ζ	Near-greedy VI		MIP	
	policy profile	average policy size		policy profile
0	 (100.0%)	1	1	(100.0%) 
0.01	 (99.0%)	1.5	1.5	(99.0%) 
0.02	 (98.1%)	1.75	1.75	(98.0%) 
0.03	 (97.1%)	2	2.25	(97.0%) 
0.04	 (96.2%)	2.25	2.5	(96.1%) 
0.05	 (95.2%)	2.75	2.75	(95.2%) 
0.1	 (90.2%)	4	4	(90.2%) 
0.2	 (90.2%)	4	4	(90.2%) 
1	 (90.2%)	4	4	(90.2%) 

Figure 9. SVPs learned by the near-greedy and MIP algorithms on Chain-5 at different ζ s. Parenthesized percentages denote the worst-case near-optimality.


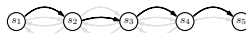
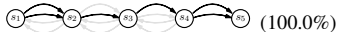
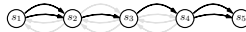

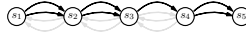

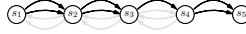







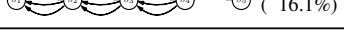
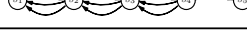
ζ	Near-greedy VI		MIP	
	policy profile	average policy size		policy profile
0	 (100.0%)	1	1	(100.0%) 
0.01	 (100.0%)	1.5	1.5	(100.0%) 
0.02	 (98.1%)	1.75	1.75	(98.9%) 
0.03	 (97.9%)	1.75	1.75	(98.9%) 
0.04	 (96.8%)	2	2	(96.8%) 
0.05	 (96.8%)	2	2	(96.8%) 
0.1	 (96.8%)	2	2	(96.8%) 
0.2	(did not converge)	-	2	(96.8%) 
1	 (16.1%)	4	4	(16.1%) 

Figure 10. SVPs learned by the near-greedy and MIP algorithms on CyclicChain-5 at different ζ s. Parenthesized percentages denote the worst-case near-optimality.

G. Policy Evaluation for SVPs

For completeness, we describe the policy evaluation algorithm for an SVP and show that the update is a contraction, thereby, guaranteeing convergence.

Given an SVP π , the value functions are defined as:

$$V^\pi(s) = \min_{a \in \pi(s)} Q^\pi(s, a)$$

$$Q^\pi(s, a) = \mathbb{E}_{r, s'} [r + \gamma V^\pi(s')]$$

The value function for any given policy π can be evaluated easily via a simple modification of iterative policy evaluation algorithm for deterministic/stochastic policies (Sutton & Barto, 2018):

Algorithm 2 Iterative policy evaluation for set-valued policies

```

1: Input: SVP  $\pi$ 
2: Initialize  $Q(s, a) = 0$  for all  $s \in \mathcal{S}, a \in \mathcal{A}$ 
3: repeat
4:    $\Delta \leftarrow 0$ 
5:   for each  $s, a \in \mathcal{S} \times \mathcal{A}$  do
6:      $Q'(s, a) = \mathbb{E}_{r, s'} [r + \gamma \min_{a' \in \pi(s')} Q(s', a')]$ 
7:      $\Delta \leftarrow \max(\Delta, |Q'(s, a) - Q(s, a)|)$ 
8:   end for
9:    $Q \leftarrow Q'$ 
10: until  $\Delta < \theta$ 
11: return  $Q$ 
    
```

We now show that the update in Algorithm 2 is a contraction:

Lemma 1. For any pair of action-value functions Q_1 and Q_2 , and a given policy π , we have:

$$\|\mathcal{T}Q_1 - \mathcal{T}Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$

Proof. For any $s, a \in \mathcal{S} \times \mathcal{A}$, we have:

$$\begin{aligned}
 |(\mathcal{T}Q_1)(s, a) - (\mathcal{T}Q_2)(s, a)| &= |\mathbb{E}_{r, s'} [r + \gamma V_1^\pi(s')] - \mathbb{E}_{r, s'} [r + \gamma V_2^\pi(s')]| \\
 &= \gamma \left| \mathbb{E}_{s'} \left[\min_{a_1 \in \pi(s')} Q_1(s', a_1) - \min_{a_2 \in \pi(s')} Q_2(s', a_2) \right] \right| \\
 &\leq \gamma \max_{s \in \mathcal{S}} \left| \left[\min_{a_1 \in \pi(s)} Q_1(s, a_1) - \min_{a_2 \in \pi(s)} Q_2(s, a_2) \right] \right| \\
 &\leq \gamma \|Q_1 - Q_2\|_\infty. \quad \square
 \end{aligned}$$

The contraction lemma further implies that Algorithm 2 converges to the unique fixed point of the value function of the policy π . As the update is a straightforward modification of the usual Bellman operator, we can implement/analyze a fitted policy evaluation algorithm for SVPs as well.

H. Clinical Task Details

Following Komorowski et al. (2018), we extracted 48 physiological features (Table 3) to represent each patient.

Table 3. The 48 physiological features

<p>Demographics/Static Source tables: PATIENTS, ADMISSIONS, ICUSTAYS, CHARTEVENTS, elixhauser_quant</p> <ul style="list-style-type: none"> • Shock Index • Elixhauser • SIRS • Gender • Re-admission • GCS - Glasgow Coma Scale • SOFA - Sequential Organ Failure Assessment • Age
<p>Lab Values Source tables: CHARTEVENTS, LABEVENTS</p> <ul style="list-style-type: none"> • Albumin • Arterial pH • Calcium • Glucose • Hemoglobin • Magnesium • PTT - Partial Thromboplastin Time • Potassium • SGPT - Serum Glutamic-Pyruvic Transaminase • Arterial Blood Gas • BUN - Blood Urea Nitrogen • Chloride • Bicarbonate • INR - International Normalized Ratio • Sodium • Arterial Lactate • CO2 • Creatinine • Ionised Calcium • PT - Prothrombin Time • Platelets Count • SGOT - Serum Glutamic-Oxaloacetic Transaminase • Total bilirubin • White Blood Cell Count
<p>Vital Signs Source tables: CHARTEVENTS</p> <ul style="list-style-type: none"> • Diastolic Blood Pressure • Systolic Blood Pressure • Mean Blood Pressure • PaCO2 • PaO2 • FiO2 • PaO/FiO2 ratio • Respiratory Rate • Temperature (Celsius) • Weight (kg) • Heart Rate • SpO2
<p>Intake and Output Events Source tables: INPUTEVENTS_CV, INPUTEVENTS_MV, OUTPUTEVENTS</p> <ul style="list-style-type: none"> • Fluid Output - 4 hourly period • Total Fluid Output • Mechanical Ventilation
<ul style="list-style-type: none"> • Timestep