

DEEP OFFLINE REINFORCEMENT LEARNING FOR REAL-WORLD TREATMENT OPTIMIZATION APPLICATIONS

A PREPRINT

Mila Nambiar^{1,*}, Supriyo Ghosh^{1,2,*}, Yu En Chan¹, Priscilla Ong¹, Yong Mong Bee³, and Pavitra Krishnaswamy¹

¹Institute for Infocomm Research (I²R), A*STAR, Singapore

²Microsoft, Bengaluru, Karnataka, India

³Department of Endocrinology, Singapore General Hospital, Singapore

*Both authors contributed equally to this research.

June 14, 2023

ABSTRACT

There is increasing interest in data-driven approaches for recommending optimal treatment strategies in many chronic disease management and critical care applications. Reinforcement learning methods are well-suited to this sequential decision-making problem, but must be trained and evaluated exclusively on retrospective medical record datasets as direct online exploration is unsafe and infeasible. Despite this requirement, the vast majority of treatment optimization studies use off-policy RL methods (e.g., Double Deep Q Networks (DDQN) or its variants) that are known to perform poorly in purely offline settings. Recent advances in offline RL, such as Conservative Q-Learning (CQL), offer a suitable alternative. But there remain challenges in adapting these approaches to real-world applications where suboptimal examples dominate the retrospective dataset and strict safety constraints need to be satisfied. In this work, we introduce a practical and theoretically grounded transition sampling approach to address action imbalance during offline RL training. We perform extensive experiments on two real-world tasks for diabetes and sepsis treatment optimization to compare performance of the proposed approach against prominent off-policy and offline RL baselines (DDQN and CQL). Across a range of principled and clinically relevant metrics, we show that our proposed approach enables substantial improvements in expected health outcomes and in accordance with relevant practice and safety guidelines.

Keywords Offline reinforcement learning, dynamic treatment optimization, sepsis, type 2 diabetes, sampling, safety constraints

1 Introduction

Deep reinforcement learning (RL) has recently experienced a surge in popularity thanks to demonstrated successes in game playing (e.g. Atari and Go [Mnih et al., 2013, Silver et al., 2017]), and with AI bots (e.g. ChatGPT¹). Given its ability to learn from large real-world experience datasets, there is also immense excitement about the potential of deep RL for clinical decision support applications. In many such applications, the objective is to leverage historical medical records containing information on patient characteristics, disease state evolution, treatment decisions and clinical outcomes; and learn treatment policies that will optimize clinical outcomes of interest. Notably, deep RL has been used for treatment optimization for a range of clinical conditions including sepsis, hypertension, type 2 diabetes, and cancer [Raghu et al., 2017a, Roggeveen et al., 2021, Sun et al., 2021, Zheng et al., 2021, Tseng et al., 2017]. However, unlike traditional game-playing or consumer-oriented applications on which deep RL methods have been developed and widely tested, treatment optimization applications are not amenable to learning through active interaction. This is due to critical safety concerns, which forbid direct online exploration of treatment alternatives on patients.

¹<https://openai.com/blog/chatgpt/>

Offline RL (also known as batch RL [Lange et al., 2012]), an approach to learn from large, previously collected datasets without any interaction with an environment, is then ideally suited for treatment optimization applications. Yet, the treatment optimization literature has almost exclusively focused on traditional value-based off-policy RL methods, particularly Double Deep Q Networks (DDQN) [van Hasselt et al., 2016] and its variants [Sun et al., 2021, Zheng et al., 2021, Raghu et al., 2017a, Lu et al., 2020, Peng et al., 2018, Yu et al., 2019, Zhu et al., 2021]. However, direct use of off-policy RL algorithms in an offline setting is known to perform poorly in general, due to issues with bootstrapping from out-of-distribution (OOD) actions [Kumar et al., 2019a] and overfitting to unseen actions [Agarwal et al., 2020, Fu et al., 2019]. In other words, off-policy methods could overestimate the Q-values of unseen state-action pairs, and mistakenly select unacceptable or even unsafe actions. Recently proposed offline RL methods such as Conservative Q-Learning (CQL) and Model-based Offline Policy Optimization (MOPO) [Kumar et al., 2020, Yu et al., 2020a] address this overestimation problem by regularizing the Q-values for unseen actions during training and by lower-bounding value function estimates.

However, translating these advances in offline RL directly to real-world treatment optimization applications remains challenging. One key challenge is that medical record datasets reflect real-world clinical practice and hence contain a mixture of both optimal and suboptimal actions. In many cases, suboptimal treatments may be prescribed due to patient preferences, communication difficulties, time and resource restrictions, limitations in clinician experience, and/or inherent uncertainty in determining the best treatment strategy (e.g., due to conflicting clinical trial evidence) [Nemati et al., 2016a, Shah et al., 2005]. Often, these practice barriers give rise to behavior policy distributions that are heavily imbalanced, with the frequency of suboptimal actions even outweighing the frequency of optimal actions. In this context, offline RL methods that severely penalize out-of-distribution actions may result in overly conservative policies.

To address this challenge, we leverage sampling methodologies to adapt offline RL methods for scenarios where suboptimal examples dominate the retrospective dataset. Specifically, our approach samples historical records of transitions (i.e., state, action, and reward tuples) corresponding to each action without altering the transition probabilities, in order to increase the proportion of less frequently seen actions in the training data. We performed extensive experiments to compare performance of a popular off-policy RL baseline (DDQN) and a SOTA offline RL method (Conservative Q Learning) with and without our sampling approach on two real-world tasks for type 2 diabetes and sepsis treatment optimization. We assessed expected health outcomes via principled off-policy evaluations and characterized consistency with relevant practice and safety guidelines for the different methods.

Our main contributions are summarized as follows:

- We demonstrate that CQL, a SOTA offline RL method, can be applied in real-world treatment optimization applications to make recommendations that are more aligned with clinical practice than DDQN, a popular off-policy RL method, while also improving expected health outcomes over DDQN and the standard of care (SoC).
- We argue theoretically and demonstrate empirically that when an intuitive heuristic to strictly enforce safety constraints during policy execution is applied to CQL’s and DDQN’s recommendations, the relative improvement of CQL over DDQN extends to constrained recommendations.
- We propose a transition sampling approach to address action imbalance, and show that this increases the likelihood of CQL selecting less frequently seen actions, while continuing to penalize value estimates for out-of-distribution actions.
- Extensive experimental results for two real-world healthcare applications demonstrate that CQL with sampling substantially improves expected health outcomes over the SoC and CQL baselines, while ensuring high alignment with clinical practice.

Our results suggest that offline RL, as opposed to off-policy RL, should be leveraged as a means of devising safe and clinically effective policies for treatment optimization problems.

2 Related Work

Our work focuses on deep offline RL methods for treatment optimization. We categorize existing relevant research into three threads: (a) Treatment optimization using RL; (b) Offline RL methods and their applications; and (c) Practical challenges associated with RL-based treatment optimization.

Treatment optimization using deep RL: The overall literature on RL for treatment optimization is comprehensively surveyed in Yu et al. [2021a]. Here, we review recent studies on deep reinforcement learning for treatment optimization applications. This literature has largely focused on optimizing management of complex syndromes in intensive inpatient

settings [Raghu et al., 2017a,b, Roggeveen et al., 2021], optimizing medication dosing in anesthesia and critical care [Schamberg et al., 2020, Yu et al., 2020b], and optimizing treatment choices for chronic diseases in outpatient settings [Tseng et al., 2017, Sun et al., 2021, Zheng et al., 2021]. Commonly, these works apply value-based off-policy deep RL algorithms such as DQN, DDQN, and variants (e.g., with dueling architecture, recurrent networks) on retrospective or batch datasets – an approach which is known to suffer from distribution shift between the learned and behavior policies and overfit to unseen actions [Raghu et al., 2017a,b, Peng et al., 2018, Yu et al., 2019, Lu et al., 2020, Zheng et al., 2021, Sun et al., 2021].

Offline RL: The need to learn optimal policies in practical data-driven decision making scenarios has led to the development of offline RL algorithms. These algorithms are set up to learn effectively from retrospective data collected under some behavior policy, without any direct exploration or environmental interaction during training. We review recent works on deep offline RL. First, implicit constraint Q-Learning [Yang et al., 2021] leverages imitation learning [Wang et al., 2018, Chen et al., 2020] to address the overfitting problem by avoiding querying OOD samples. Second, value-based and policy-based offline RL algorithms (e.g., CQL [Kumar et al., 2020], MOPO [Yu et al., 2020a], UWAC [Wu et al., 2021], Fisher-BRC [Kostrikov et al., 2021], COMBO [Yu et al., 2021b]) prevent over-optimism by penalizing the learned value function for OOD actions with regularization during training. A third set of offline RL algorithms (e.g., BCQ [Fujimoto et al., 2019], BEAR [Kumar et al., 2019b], BRAC [Wu et al., 2019]) uses regularization to penalize deviations between the learned and behaviour policies. A few recent studies have explored offline RL algorithms for treatment optimization [Fatemi et al., 2022, 2021, Killian et al., 2020]. Among these, Fatemi *et al.* [Fatemi et al., 2022] proposed a modification to BCQ [Fujimoto et al., 2019] for a continuous-time semi-MDP setting. Further, Fatemi *et al.* [Fatemi et al., 2021] proposed a method to identify states from which negative outcomes are unavoidable. Finally, Killian *et al.* [Killian et al., 2020] used state representation learning with discretized BCQ. However, translation of advances in offline RL to real-world treatment optimization applications is still nascent, and several practical challenges remain.

Practical challenges in RL-based treatment optimization: A key challenge in real-world clinical applications stems from action imbalance due to dominance of suboptimal, and often conservative, actions in the data. The offline RL community is starting to recognize that overly conservative policies impede generalization and performance, and attempts to reduce conservativeness of CQL and variants are emerging. For example, the very recently proposed MCQ [Lyu et al., 2022] adapts CQL to actively train OOD actions. However, such strategies are not designed to directly address action imbalance in the retrospective data and cannot be generalized across offline RL methods. Inspired by sampling-based approaches to handle class imbalance in supervised learning [Kubat and Matwin, 1997, Ling and Li, 1998, Schistad Solberg and Solberg, 1996, Chawla et al., 2002, He et al., 2008], we propose a transition sampling scheme to address action imbalance and demonstrate its ability to improve quality and relevance of the resulting treatment policies.

3 Background

In this section, we represent the treatment optimization problem within an RL framework and introduce baseline approaches such as Q-learning and DDQN.

3.1 Problem Formulation

We consider a setting where the patient state evolves according to an underlying Markov Decision Process (MDP). This MDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$, where \mathcal{S} denotes the set of all possible states, \mathcal{A} denotes the set of discrete permissible actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbf{P}(\mathcal{S})$ represents the transition function providing the next-state distribution after executing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes a reward function providing the expected immediate reward for executing action a in state s , and $\gamma \in [0, 1]$ is a discount factor. Let T_i denote the treatment horizon length for patient i . At time t , for patient i , a clinician observes patient state $s_{i,t} \in \mathcal{S}$ and recommends a treatment, or action $a_{i,t}$ from a finite and discrete action set $\mathcal{A} = \{1, 2, \dots, A\}$. The reward $r(s, a)$ is increasing in positive health outcomes (e.g., lab results within target range) and decreasing in negative health outcomes (e.g., mortality). The goal is to identify a treatment policy $\pi : \mathcal{S} \rightarrow \mathbf{P}(\mathcal{A})$ that chooses the action at each time step t that maximizes expected cumulative reward as follows:

$$\max_{\pi} \mathbb{E} \left[\sum_{t'=t}^{T_i} \gamma^{t'-t} [r(s_{t',i}, \pi(s_{t',i}))] \right] \quad (1)$$

For some medical conditions, e.g., type 2 diabetes, feasible treatments are constrained by clinical practice and safety guidelines. This setting can be modeled as a constrained optimization problem:

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E} \left[\sum_{t'=t}^{T_i} \gamma^{t'-t} [r(s_{t',i}, \pi(s_{t',i}))] \right] \\ \text{s.t.} \quad & \mathbb{E}[C_j(s_{t,i}, \pi(s_{t',i}))] \leq c_j \quad \forall j \in \{1, \dots, C\} \end{aligned} \quad (2)$$

$C_j(s_{t,i}, \pi(s_{t',i}))$ denotes the type j constraint violation cost at time t for taking action $\pi(s_{t',i})$ at state $s_{t,i}$. c_j is the threshold on cumulative constraint violation cost over the entire horizon for type j constraints. It can be set as 0 for treatment recommendation applications, to represent strict safety guidelines.

We are specifically interested in the offline reinforcement learning setting, where the policy that solves Equation 1 must be learned solely from some retrospective dataset $\mathcal{D} = \{(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t+1}), \text{ for } i = 1, \dots, I; t = 1, \dots, T_i\}$ generated by a behavior policy $\pi_{\beta}(s)$. This behavior policy corresponds to the SoC, and need not be optimal or strictly satisfy constraints due to real-world clinical practice challenges. As we are interested in comparing the performance of offline RL methods with off-policy RL methods, we begin by introducing the off-policy RL method, DDQN, which leverages the canonical RL algorithm, Q-learning.

3.2 Q-learning

Our goal is to learn a (potentially randomized) policy $\pi : \mathcal{S} \rightarrow \mathbf{P}(\mathcal{A})$ that specifies the action to take in each state $s \in \mathcal{S}$. The value at state s with respect to policy π is estimated as the expected cumulative reward for executing π :

$$V^{\pi}(s) = \mathbb{E}_{\pi} \left[\sum_{t=0}^T \gamma^t r(s_t, \pi(s_t)) | s_0 = s \right] \quad (3)$$

The goal is to learn an optimal policy π^* that maximizes $V^{\pi}(s)$ for all s . In a finite MDP setting, tabular Q-learning can asymptotically learn π^* by learning the optimal Q function [Watkins, 1989] Q^{π^*} , where Q^{π^*} is obtained using the following Bellman equation:

$$Q^{\pi^*}(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \mathbf{P}(s'|s, a)} \left[\max_{a'} Q^{\pi^*}(s', a') \right] \quad (4)$$

If Q^{π^*} is known, the optimal action a^* for a given state s is computed by $a^* = \pi^*(s) = \arg \max_a Q^{\pi^*}(s, a)$. Q^{π^*} can be estimated by iteratively applying the following Bellman operator:

$$\mathcal{B}^k Q^k(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \mathbf{P}(s'|s, a)} \left[\max_{a'} Q^k(s', a') \right] \quad \forall k \in \{1, 2, \dots\} \quad (5)$$

Further, each Q^k can be represented using a neural network $N(s, a; \theta_k)$ with input s , output of dimension A , and parameters θ_k , which are iteratively updated during training.

3.3 DDQN

DDQN approximates Q values using two neural networks: A master network and a target network. At every iteration k , the parameters of the master network are updated using Equation 5, where the inner maximization is replaced with the following function approximator:

$$\max_{a'} Q^k(s, a') = N(s, \arg \max_{a'} N(s, a'; \theta_k); \theta'_k). \quad (6)$$

The target network is then updated by setting $\theta'_k = \theta_k$ every τ iterations for some hyperparameter τ . By using different networks to select and evaluate actions, DDQN attempts to address the issue of Q value overestimation that may arise from taking the maximum of the estimated Q values in Equation 5 [van Hasselt et al., 2016].

4 Methods

In this work, we employ offline RL methods to improve the quality of treatment recommendations. We first describe a SOTA offline RL algorithm, CQL, that is used to generate a recommendation policy. Then, we explain our methodology to address the challenge of training CQL on retrospective datasets with action imbalance. Finally, we describe an intuitive heuristic to enforce strict constraint satisfaction, and discuss how this is expected to impact the performance of CQL, or more generally of offline RL methods.

4.1 Conservative Q-Learning (CQL)

A critical issue with directly applying Equation 5 in the offline RL setting is that the Q values of OOD actions may be overestimated. To mitigate this, CQL combines the standard Q-learning objective of solving the Bellman equation with an additional regularization term that minimizes Q values for OOD actions (i.e., large Q values for OOD actions are penalized) [Kumar et al., 2020]. This gives rise to the objective

$$\min_Q \max_{\mu} \alpha \cdot (\mathbb{E}_{s \sim \mathcal{D}, a \sim \mu(a|s)} [Q(s, a)] - E_{s \sim \mathcal{D}, a \sim \pi_{\beta}(a|s)} [Q(s, a)]) + \frac{1}{2} \mathbb{E}_{(s, a, s') \sim \mathcal{D}} [(Q(s, a) - \mathcal{B}^k Q^k(s, a))^2] + \mathcal{R}(\mu). \quad (7)$$

Here, μ is some distribution over the actions conditional on state, and $\mathcal{R}(\mu)$ is a regularization term. Alternatives for $\mathcal{R}(\mu)$ proposed in Kumar et al. [2020] include: (i) Entropy, (ii) the negative of the KL divergence between μ and some given distribution, e.g., the learned policy at the previous iteration during model training, and (iii) a variant inspired by distributionally robust optimization, which penalizes the variance in the Q -function across actions and states.

Finally, $\alpha : \alpha > 0$ is a hyperparameter controlling the trade-off between the two objectives. While α can be tuned to learn less conservative policies, this alone may be insufficient to address the impact of severe action imbalance in the retrospective dataset \mathcal{D} . Hence, we propose a sampling method to reduce action imbalance.

4.2 CQL with sampling

CQL is designed to ensure that Q value estimates for OOD actions are conservative. In settings like treatment optimization where the behavior policy exhibits large action imbalance, this may result in CQL predominantly recommending actions that are frequently recommended by the behavior policy, even when this is suboptimal. We thus propose to apply sampling approaches, which have been successfully used in many real-world classification problems to address class imbalance [Kubat and Matwin, 1997, Schistad Solberg and Solberg, 1996, Ling and Li, 1998, He et al., 2008, Chawla et al., 2002], to similarly address the problem of action imbalance in offline RL.

Consider the following sampling procedure: Given action a , $1 \leq a \leq A$, suppose we sample with replacement a dataset $\hat{\mathcal{D}}$ from the set of historical transitions $\{(s_t, a, s_{t+1})\}$. Denote the ratio of proportions of this set after sampling to before sampling as w_a ($w_a > 1$: Action a has been used less frequently in the retrospective dataset). Then $\hat{\mathcal{D}}$ has distribution

$$\Pr_{\hat{\mathcal{D}}}[a|s_t] = w_a \Pr_{\mathcal{D}}[a|s_t]. \quad (8)$$

Since $\hat{\mathcal{D}}$ is created by sampling with replacement from the set of historical transitions with action a , we also have, for each a :

$$\Pr_{\hat{\mathcal{D}}}[s_{t+1}|s_t, a] = \Pr_{\mathcal{D}}[s_{t+1}|s_t, a], \quad (9)$$

i.e., the transition probabilities are the same for \mathcal{D} and $\hat{\mathcal{D}}$. In the following, we will denote $\Pr_{\hat{\mathcal{D}}}[a|s_t]$ as $\pi_{\hat{\beta}}$. To see how this approach affects the CQL recommendations, we write the CQL objective (Equation 7) under the distribution $\hat{\mathcal{D}}$:

$$\begin{aligned} & \mathbb{E}_{s_t \sim \hat{\mathcal{D}}} \left[\alpha \cdot (\mathbb{E}_{a_t \sim \mu(a_t|s_t)} [Q(s_t, a_t)] - \mathbb{E}_{a_t \sim \pi_{\hat{\beta}}(a_t|s_t)} [Q(s_t, a_t)]) \right. \\ & \quad \left. + \frac{1}{2} \mathbb{E}_{(a_t, s_{t+1}) \sim \hat{\mathcal{D}}|s_t} [(Q(s_t, a_t) - \mathcal{B}^k Q^k(s_t, a_t))^2] \right] + \mathcal{R}(\mu) \\ = & \mathbb{E}_{s_t \sim \hat{\mathcal{D}}} \left[\sum_a \alpha \cdot Q(s_t, a) (\mu(a|s_t) - w_a \pi_{\beta}(a|s_t)) \right. \\ & \quad \left. + w_a \pi_{\beta}(a|s_t) \mathbb{E}_{s_{t+1} \sim \mathcal{D}|s_t, a} [(Q(s_t, a) - \mathcal{B}^k Q^k(s_t, a))^2] \right] + \mathcal{R}(\mu). \end{aligned} \quad (10)$$

Here, the expectation is taken over the state s_t under $\hat{\mathcal{D}}$. Note that $s_t \sim \mathcal{D}$ is distinct from $s_t \sim \hat{\mathcal{D}}$ as the frequency of transitions with state s_t is the sum of the frequencies of transitions with state s_t and action a across all a ; This differs from \mathcal{D} to $\hat{\mathcal{D}}$ due to Equation 8. The right hand side of Equation 10 then follows from our assumptions about the sampling process, where the first summand in the expectation follows from Equation 8 and the second summand is derived from Equation 9. Thus when $w_a > 1$, the CQL overestimation term (first summand) decreases with sampling compared to without sampling, while the Bellman loss term (second summand) receives greater weight after sampling is applied, i.e., CQL becomes less conservative with respect to action a . Conversely, when $w_a < 1$, the CQL overestimation term (first summand) increases with sampling compared to without sampling, while the Bellman

loss term (second summand) receives smaller weight after sampling is applied, i.e., CQL becomes more conservative with respect to action a .

Motivated by this analysis, we propose combining CQL with sampling as follows: Instead of adopting A hyperparameters w_a , which could be computationally costly to tune, we introduce a single hyperparameter K . By tuning K through grid-search, we can in turn tune all w_a to reduce action imbalance. Denoting the mean number of transitions per action in \mathcal{D} as σ , we have: 1) Undersampling - for each action with more than $K\sigma$ transitions, sample $K\sigma$ transitions with replacement; 2) Oversampling - for each action with fewer than $K\sigma$ transitions, sample $K\sigma$ transitions with replacement; and 3) Under+oversampling - for each action, sample σ transitions with replacement. We then trained CQL on the sampled datasets.

4.3 Constraint satisfaction

The constraints from the diabetes setting take the form

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\pi} \left[\sum_{t=0}^T \gamma^t R(s_t, a_t) \mid s_0, a_0 = 0 \right] \\ \text{s.t.} \quad & \pi(s_t) \in A_F(s_t). \end{aligned} \quad (11)$$

For constrained optimization problems (see Equation 2), direct application of CQL and DDQN does not guarantee constraint satisfaction during policy execution. For this special case, the learned policies of any value-based RL agent can be easily adapted to ensure strict constraint satisfaction. Given state s_t , denote the feasible set of actions as $A_F(s_t)$. For a given value-based RL agent π_{RL} , define the corresponding constrained policy as

$$\pi_{RL,c}(s_t) = \arg \max_{a \in A_F(s_t)} Q^{\pi_{RL}}(s_t, a). \quad (12)$$

Here, $Q^{\pi_{RL}}(s_t, a)$ denotes the RL agent’s estimated Q value. $\pi_{RL,c}(s_t)$ thus recommends the feasible action with the highest predicted Q value. Then, the constrained and unconstrained recommended actions are the same when the latter is feasible.

Intuitively, this means that if the RL agent’s rate of constraint satisfaction is high, the optimality gap for the constrained recommendations should be close to the optimality gap for the unconstrained recommendations. This is precisely expected to be the case for offline RL algorithms in treatment optimization settings. Since the SoC’s rate of constraint satisfaction should be high (the SoC reflects domain experts’ attention to safety considerations), and the offline RL agent’s recommendations should not deviate too much from the SoC recommendations, the latter’s constraint satisfaction rate should be high as well. Any performance guarantees for the offline RL agents’ unconstrained recommendations should continue to hold for constrained recommendations. This observation is supported by Property 1 below (proof in Appendix A.1).

Property 1. *Let π^* be an unconstrained policy solving Equation 1, and let π_c^* be a constrained policy solving Equation 11. Assuming that the reward $r(s, a)$ is bounded as $\underline{r} \leq r(s, a) \leq \bar{r}$, the optimality gap for $\pi_{RL,c}(s_t)$ can be bounded in terms of the optimality gap for $\pi_{RL}(s_t)$ as*

$$\mathbb{E}[V^{\pi_c^*}(s_0)] - \mathbb{E}[V^{\pi_{RL,c}}(s_0)] \leq \frac{\bar{r} - \underline{r}}{1 - \gamma} \sum_{t=0}^T \Pr[\exists t \text{ s.t. } \pi(s_t) \notin A_F(s_t)] + \mathbb{E}[V^{\pi^*}(s_0)] - \mathbb{E}[V^{\pi_{RL}}(s_0)]. \quad (13)$$

5 Experimental Design

5.1 Treatment optimization applications

We conducted experiments comparing the recommendations of the different RL agents for two clinical applications: 1) type 2 diabetes, and 2) sepsis. We detail the tasks and datasets for these below.

5.1.1 Diabetes treatment recommendation

Task. We considered the problem of recommending antidiabetic treatment regimens to type 2 diabetes patients in an outpatient setting. At each visit, the doctor prescribes a treatment regimen from among 13 options: 1) Maintain, 2) increase, or 3) decrease the dosages of previously prescribed drugs, or start a new drug from among the subgroups 4) acarbose, 5) DPP-4 inhibitors, 6) biguanides, 7) SGLT2 inhibitors, 8) sulphonylureas, 9) thiazolidinediones, 10) GLP-1 RAs, 11) long or intermediate acting insulins, 12) premixed insulins, and 13) rapid acting insulins. The goal is to treat

the patient's glycaeted haemoglobin (HbA_{1c}) down to a target of 7%, minimize the incidence of severe hypoglycemia, and reduce the incidence of complications such as heart failure [Association, 2022]. This results in the following expression for reward at patient i 's visit at time t

$$r_{i,t} = 1\{s_{i,t+1}^{HbA_{1c}} \leq 7\} - 2s_{i,t+1}^{Hypo} - 4s_{i,t+1}^{Compl} \quad (14)$$

where $s_{i,t}^{HbA_{1c}}$ is the HbA_{1c}, $s_{i,t}^{Hypo}$ indicates the occurrence of hypoglycemia, and $s_{i,t}^{Compl}$ indicates the occurrence of complications or death at time t . Similar reward functions are used in Sun et al. [2021] and Zheng et al. [2021]. For safety reasons, the treatment recommendation must also adhere strictly to clinical guidelines:

$$C1: A_F^1(s_{i,t}) = \mathcal{A}_D \setminus \{6\} \text{ if } s_{i,t}^{eGFR} < 30 \text{ml/min/1.73m}^2 \quad (15)$$

$$C2: A_F^2(s_{i,t}) = \mathcal{A}_D \setminus \{7\} \text{ if } s_{i,t}^{eGFR} < 45 \text{ml/min/1.73m}^2 \quad (16)$$

$$C3: A_F^3(s_{i,t}) = \mathcal{A}_D \setminus \{10\} \text{ if } s_{i,t}^{pancr} = 1 \quad (17)$$

$$C4: A_F^4(s_{i,t}) = \mathcal{A}_D \setminus \{8\} \text{ if } s_{i,t}^{age} > 70, \quad (18)$$

where $s_{i,t}^{eGFR}$ is the estimated glomerular filtration rate (eGFR), $s_{i,t}^{pancr}$ is the incidence of pancreatitis, $s_{i,t}^{age}$ is the patient's age, and $\mathcal{A}_D = \{1, 2, \dots, 13\}$ is the set of actions. The feasible set $A_F(s_{i,t})$ is then

$$A_F(s_{i,t}) = \bigcap_{j=1}^4 A_F(s_{i,t}^j). \quad (19)$$

Data. We studied this problem using electronic medical records from outpatient prescription visits for type 2 diabetes patients within the Singapore Diabetes Registry [Lim et al., 2021]. Our study was approved by the relevant Institutional Review Board with a waiver of informed consent. For each patient visit at time t , the state was defined by 55 variables describing the patient medical profile, including demographics (age, gender, ethnicity), physical measurements (heart rate, blood pressure, BMI), blood and urine laboratory data (HbA_{1c}, fasting glucose, lipid panel, full blood counts, creatinine, estimated glomerular filtration rate, urine albumin/creatinine ratio), medical history (diabetes duration, utilization details, comorbidities and complications), and the previous visit prescription. We included prescription visits based on two inclusion criteria: (a) visit had at least one preceding prescription, and (b) visit had at least 1 HbA_{1c} measurement within the past month and at least 1 eGFR reading within the past year. This yielded 1,302,461 visits for 71,863 patients.

Among these visits, we observe significant action imbalance. The most and least common action account for 64.0% and 0.01% of visits respectively. The most common action is "No change" and is prescribed in 51.2% of visits where the patient's HbA_{1c} is above the target of 7%. This could be indicative of clinical inertia, where treatments are not intensified appropriately due to lack of time during the consultation, or lack of expertise in primary care settings [Shah et al., 2005]. The action imbalance observed in this dataset, along with an over-representation of suboptimal actions, suggests that it is a good candidate for the methods proposed in Section 4.2.

5.1.2 Sepsis treatment recommendation

Task. We also considered the problem of treating ICU patients with sepsis. At each 4 hour window t during patient i 's ICU stay, the clinician administers fluids and/or vasopressors, each discretized into 5 volumetric categories. The treatment is described by the tuple (u, v) , $1 \leq u, v \leq 5$, with 25 treatment options in total. The goal being to prevent patient mortality, we model the treatment optimization problem with the following reward function:

$$r_{i,t} = \begin{cases} 0 & \text{if } t < T \\ 1 & \text{if } t = T \text{ and } s_{i,t}^{Mort} = 0 \\ -1 & \text{if } t = T \text{ and } s_{i,t}^{Mort} = 1 \end{cases} \quad (20)$$

where $s_{i,t}^{Mort}$ is the incidence of mortality within a 48 hour window of time t . Similar reward functions have been used in Fatemi et al. [2021] and Komorowski et al. [2018].

Data. We studied this problem using data on a cohort of sepsis patients generated from the publicly available MIMIC (Medical Information Mart for Intensive Care) - III dataset [Johnson et al., 2016] following Fatemi et al. [2021]². For each ICU stay and 4 hour window t , the state was defined by 44 variables describing the patient medical profile, including demographics (age, gender), physical measurements (heart rate, blood pressure, weight), and blood and

²We directly applied the associated code base provided in https://github.com/microsoft/mimic_sepsis.

urine laboratory data (glucose, creatinine). There were a total of 18923 unique ICU stays. We observed large action imbalance, with the most and least common actions accounting for 27.1% and 1.9% of visits respectively. The most common action was (1, 1), corresponding to the lowest possible dose ranges for IV fluids and vasopressors. Applying existing offline RL methods to learn from this dataset may thus result in recommendations for insufficiently intensive treatments.

5.2 Evaluations

In the absence of data on counterfactuals, we considered evaluation techniques that use retrospective data.

5.2.1 Weighted Importance Sampling (WIS)

We applied WIS, an off-policy evaluation technique widely used in the treatment optimization literature [Komorowski et al., 2018, Raghu et al., 2017a, Roggeveen et al., 2021, Raghu et al., 2017b, Peng et al., 2018], to estimate the value of each RL agent’s policy. We define the **WIS score** for an RL agent as follows. First, we define the importance ratios $\rho_{i,t}$ as the ratio of the likelihood of the RL agent policy π_{RL} and the likelihood of the SoC policy π_{Clin} selecting the SoC action $a_{i,t}$ given state $s_{i,t}$. We then define trajectory-wise WIS estimators V_i^{WIS} for trajectory $i, i = 1, \dots, N$, in terms of these importance ratios, and average V_i^{WIS} across trajectories:

$$\begin{aligned} \rho_{i,t} &= \pi_{RL}(a_{i,t}|s_{i,t}) / \pi_{Clin}(a_{i,t}|s_{i,t}) \\ \rho_{i,1:t} &= \prod_{t'=1}^t \rho_{i,t'}; \quad w_{i,t} = \sum_{i=1}^N \rho_{i,1:t} / N \\ V_i^{WIS} &= \frac{\rho_{i,1:t}}{w_{i,T}} \cdot \sum_{t=1}^T \gamma^{t-1} r_{i,t}; \quad WIS = \frac{1}{N} \sum_{i=1}^N V_i^{WIS} \end{aligned} \quad (21)$$

This gives a biased but consistent estimator of the expected cumulative reward under the RL agent’s policy [Hesterberg, 1995].

Following Komorowski et al. [2018], we estimated π_{Clin} by training a multinomial logistic regression model with the one-hot-encoded selected action as the output, and the state as the features, then by taking the predicted probabilities for the different actions. We estimated π_{RL} by "softening" the RL agent policy, i.e., approximating it with a random policy π_{RL}^{WIS} that selects from the non-optimal actions uniformly at random with some small probability. Specifically,

$$\pi_{RL}^{WIS}(s_{i,t}) = \begin{cases} a \text{ w.p. } \epsilon & \text{if } a_{i,t}^* \neq a \\ a_{i,t}^* \text{ w.p. } 1 - \epsilon & \text{otherwise.} \end{cases} \quad (22)$$

where ϵ ($0 < \epsilon \ll 1$) is a softening probability. To ensure a fair comparison between the SoC and the RL agents, we applied softening to the SoC policy and calculated WIS estimates.

5.2.2 Additional metrics.

We defined metrics of how well the RL agents’ recommendations were aligned with clinical practice.

Model Concordance Rate. This is the fraction of visits where the RL policy’s recommendation matches the SoC [Sun et al., 2021, Lin et al., 2018, Nemati et al., 2016b]:

$$\text{Model Concordance Rate} = \sum_{i,t} \mathbb{1}\{\pi_{SoC}(s_{i,t}) = \pi_{RL}(s_{i,t})\} / \sum_i T_i. \quad (23)$$

Appropriate Intensification Rate. For the type 2 diabetes application, this is the fraction, out of visits with HbA1c over 7.0%, where the RL agent recommends treatment intensification (i.e., increase dose or add new drug):

$$\text{Appr. Intens. Rate} = \sum_{i,t} \frac{\mathbb{1}\{s_{i,t}^{HbA1c} > 7.0\%, \pi_{RL}(s_{i,t}) \notin \{0, 2\}\}}{\mathbb{1}\{s_{i,t}^{HbA1c} > 7.0\%}}. \quad (24)$$

Constraint Satisfaction Rate (CSR). For the diabetes application, we defined the CSR for each constraint $j, j = 1$ to 4 as the fraction of all visits where constraint j applies such that the RL agent satisfies constraint j :

$$\text{CSR for constraint } j = \sum_{i,t} \mathbb{1}\{\pi_{RL}(s_{i,t}) \in A_F^j(s_{i,t})\} / \sum_i T_i.$$

As an illustrative example, for constraint 1, the CSR is the fraction, out of visits with eGFR under $30\text{ml}/\text{min}/1.73\text{m}^2$, where the recommendation is not "add metformin."

Table 1: $\min_a\{w_a\}$ and $\max_a\{w_a\}$ with different sampling approaches for the diabetes and sepsis applications. For undersampling and oversampling, ranges across various K are provided.

Application	Sampling approach	$\min_a\{w_a\}$	$\max_a\{w_a\}$
Diabetes	Undersampling	0.27-0.39	2.73-5.52
	Oversampling	0.66-0.82	254.89-406.87
	Under+oversampling	0.12	776.37
Sepsis	Undersampling	0.29-0.50	2.60-4.48
	Oversampling	0.67-0.85	8.05-12.67
	Under+oversampling	0.16	23.69

5.3 Implementation details

We set three seeds, and for each seed, randomly split the patients in each dataset into training, validation, and test sets with the ratio 60:20:20. We then used the `d3rlpy` library³ in Python to train CQL and DDQN. For CQL, `d3rlpy` sets the regularization term $\mathcal{R}(\mu)$ in Equation 7 as entropy, i.e., $\mathcal{R}(\mu) = \mathbb{E}[-\log(\mu)]$. This variant was also shown through ablation studies in Kumar et al. [2020] to generally outperform the other proposed variants.

For each seed, we trained (i) CQL with $\alpha \in [0.1, 0.5, 0.8, 0.9, 1.0]$; (ii) CQL with undersampling and $K \in [0.4, 0.8, 1.2]$, CQL with oversampling and $K \in [0.4, 0.8]$, and CQL with under+oversampling; and (iii) DDQN. Transition sampling was applied to the training set but not the validation and test sets. These settings resulted in the sampling weights $w_a : 1 \leq a \leq A$ given in Table 1 (Recall from Section 4.2 that w_a denotes the frequency of transitions with action a after sampling compared to before sampling). For both the diabetes and sepsis applications, under+oversampling gives rise to the greatest decrease in frequency (smallest w_a) and the greatest increase in frequency (largest w_a). Under+oversampling thus seems to be the most aggressive sampling approach for these settings, and undersampling the least aggressive approach. Finally, we applied stratified random sampling to increase sample representativeness and to ensure that the empirical distributions of the sampled datasets are close to $\hat{\mathcal{D}}$ [Teddlie and Yu, 2007]. Details are in Appendix B.2.

For each RL agent, we used multilayer perceptron architectures for the master and target Q networks and considered 2 different configurations: (i) 2 linear layers with 256 hidden units, (iii) 3 linear layers with 512 hidden units. We set the batch size to 64, learning rate to $6.25e-5$, and the target update interval to 8000 steps. We used grid-search across all hyperparameter combinations to select the model with the highest WIS score on the validation sets, and evaluated this model. To compute WIS scores, we trained multinomial logistic regression models to approximate the clinician policy (details in Appendix B.1), and applied a softening factor of 0.99. We also calculated WIS scores for the SoC using Equation 22.

6 Results and Discussion

Results for the diabetes application are in Fig. 1, Table 2, Table 3, while results for the sepsis application are in Fig. 2 and Table 4. We organize our findings into three subtopics: (i) Comparison of the offline RL method CQL with the off-policy RL method DDQN; (ii) Effect of sampling and regularization hyperparameter tuning on the performance of CQL; and (iii) Performance of the RL methods after the constraint satisfaction heuristic (Equation 12) is applied.

6.1 Comparing CQL and DDQN

Comparing the unconstrained recommendations of CQL with $\alpha = 1.0$ and DDQN for the diabetes treatment application, we first see from Fig. 1 that the distribution across the treatment options is much closer to the SoC under CQL than under DDQN. In particular, CQL is far more likely than DDQN to recommend no change, increasing dose, or decreasing dose, and far less likely than DDQN to recommend adding new medications. The model concordance rates in Table 2 support this picture. Across 3 seeds, CQL exhibits a mean model concordance rate of 62.5% with the SoC’s recommendations, compared to only 1.5% under DDQN. As a result of the greater model concordance of the SoC with CQL than with DDQN, we see from Table 3 that across the four constraints, the constraint satisfaction rates are higher under CQL (between 98.7% and 99.7%) than under DDQN (between 88.1% and 94.3%); and are closer to the constraint satisfaction rates under the SoC (between 98.1% and 100.0%). Thus CQL generates recommendations that have far greater alignment to clinical practice than DDQN.

³<https://github.com/takuseno/d3rlpy>

Table 2: Comparison of treatment recommendations by the SoC, and the unconstrained and constrained recommendations by the different RL agents, for the diabetes application. The mean and standard deviations of the results across 3 seeds are reported. For each metric, the highest mean value across the RL agents is bolded, along with the associated standard deviation.

Agent	Model Concordance Rate		WIS score		Appr. Intensification Rate	
	Unconstrained	Constrained	Unconstrained	Constrained	Unconstrained	Constrained
SoC	100.0% \pm 0.0%	N.A.	-6.741 \pm 3.549	N.A.	40.3% \pm 0.1%	N.A.
CQL with $\alpha = 1.0$	62.5% \pm 0.3%	62.6% \pm 0.3%	3.653 \pm 0.983	3.698 \pm 0.947	37.8% \pm 0.7%	37.4% \pm 0.8%
CQL with $\alpha = 0.9$	63.0% \pm 0.1%	63.1% \pm 0.1%	3.489 \pm 1.647	3.522 \pm 1.590	38.0% \pm 2.4%	37.7% \pm 2.4%
CQL with $\alpha = 0.8$	63.3% \pm 0.2%	63.3% \pm 0.2%	3.795 \pm 1.642	3.798 \pm 1.635	39.1% \pm 1.0%	38.8% \pm 1.0%
CQL with $\alpha = 0.5$	64.3% \pm 0.3%	64.3% \pm 0.3%	3.580 \pm 0.820	3.663 \pm 0.695	39.9% \pm 2.1%	39.6% \pm 2.0%
CQL with $\alpha = 0.1$	50.7% \pm 0.2%	50.9% \pm 0.2%	5.530 \pm 1.018	5.496 \pm 0.963	57.5% \pm 4.1%	57.1% \pm 4.0%
CQL with undersampling	43.6% \pm 0.2%	44.1% \pm 0.2%	5.082 \pm 1.062	5.763 \pm 1.235	73.5% \pm 3.2%	66.8% \pm 5.9%
CQL with oversampling	61.5% \pm 0.5%	61.7% \pm 0.4%	4.070 \pm 1.374	4.049 \pm 1.342	40.8% \pm 1.6%	40.3% \pm 1.7%
CQL with under+oversampling	41.3% \pm 0.5%	42.0% \pm 0.5%	5.721 \pm 0.392	4.668 \pm 1.692	76.1% \pm 10.1%	68.5% \pm 8.9%
DDQN	1.5% \pm 0.0%	1.6% \pm 0.1%	-3.055 \pm 1.577	-0.190 \pm 3.152	97.5% \pm 0.3%	97.1% \pm 0.3%

Table 3: Mean and standard deviations across 3 seeds of the CSRs for the four constraints (Equations 15-18) in the diabetes application. For each constraint, we bold the highest mean CSR(s) across RL agents, and the associated standard deviation(s).

Agent	Constraint Satisfaction Rate (CSR)			
	C1	C2	C3	C4
SoC	99.6% \pm 0.1%	99.8% \pm 0.1%	100.0% \pm 0.0%	98.1% \pm 0.0%
CQL with $\alpha = 1.0$	99.3% \pm 0.1%	99.6% \pm 0.0%	99.7% \pm 0.2%	98.7% \pm 0.2%
CQL with $\alpha = 0.9$	99.4% \pm 0.2%	99.7% \pm 0.1%	99.9% \pm 0.1%	98.7% \pm 0.1%
CQL with $\alpha = 0.8$	99.4% \pm 0.1%	99.7% \pm 0.1%	100.0% \pm 0.0%	98.8% \pm 0.1%
CQL with $\alpha = 0.5$	99.5% \pm 0.1%	99.8% \pm 0.1%	99.9% \pm 0.1%	98.9% \pm 0.2%
CQL with $\alpha = 0.1$	98.4% \pm 0.3%	99.3% \pm 0.1%	99.5% \pm 0.6%	98.4% \pm 0.2%
CQL with undersampling	99.0% \pm 0.1%	99.4% \pm 0.1%	100.0% \pm 0.0%	95.4% \pm 0.0%
CQL with oversampling	99.2% \pm 0.4%	99.6% \pm 0.1%	99.9% \pm 0.2%	98.2% \pm 0.3%
CQL with under+oversampling	97.3% \pm 0.5%	98.8% \pm 0.0%	99.9% \pm 0.1%	93.1% \pm 1.2%
DDQN	90.0% \pm 2.9%	88.1% \pm 3.0%	94.3% \pm 2.6%	93.1% \pm 0.7%

In terms of impact on health outcomes, Table 2 shows that the WIS score averaged across 3 seeds is higher under CQL (3.653) than under DDQN (-3.055). Both CQL and DDQN achieve greater WIS scores than the SoC (-6.741). We can then conclude from Equation 14 that unlike the DDQN and SoC recommendations, the CQL recommendations are not expected to lead to complications and/or mortality on average.

6.2 Effect of sampling and regularization hyperparameter tuning

Next, we compare the recommendations of the various CQL agents for the diabetes and sepsis applications. For sampling, results correspond to the agent with K selected via grid-search. Appendix C provides a sensitivity analysis of how performance depends on K .

Diabetes treatment application. Sampling and lowering α lead to greater divergence between the CQL and SoC recommendations: Both contribute to a drop in the frequency of the no change recommendation and increases in the frequencies of each of the remaining treatment options (Fig. 1). Similarly, the model concordance rate under CQL, as averaged over 3 seeds, decreases from 62.5% with $\alpha = 1.0$ to 50.7% with $\alpha = 0.1$, and to 43.6% and 41.3% with undersampling and under+oversampling respectively (Table 2). Thus both regularization hyperparameter tuning and sampling have the expected effects of reducing the action imbalance of CQL’s prescribed treatments. At the same time, the CQL agents’ recommendations are still more similar to the SoC than the DDQN baseline, i.e. as suggested in Section 4.2, hyperparameter tuning and sampling continue to encourage conservativeness. This translates to generally higher rates of constraint satisfaction than under DDQN: Constraint satisfaction rates are between 98.4% to 100.0% for CQL with $\alpha < 1$, and between 93.1% to 100.0% for CQL with sampling (Table 3).

In terms of the optimality of the recommendations with hyperparameter tuning and sampling, Table 2 shows an increase in the rate of appropriate treatment intensification from 37.8% for CQL with $\alpha = 1.0$ to 57.5% with $\alpha = 0.1$, to 73.5% with undersampling, to 40.8% with oversampling, and to 76.1% with under+oversampling. Compared to CQL

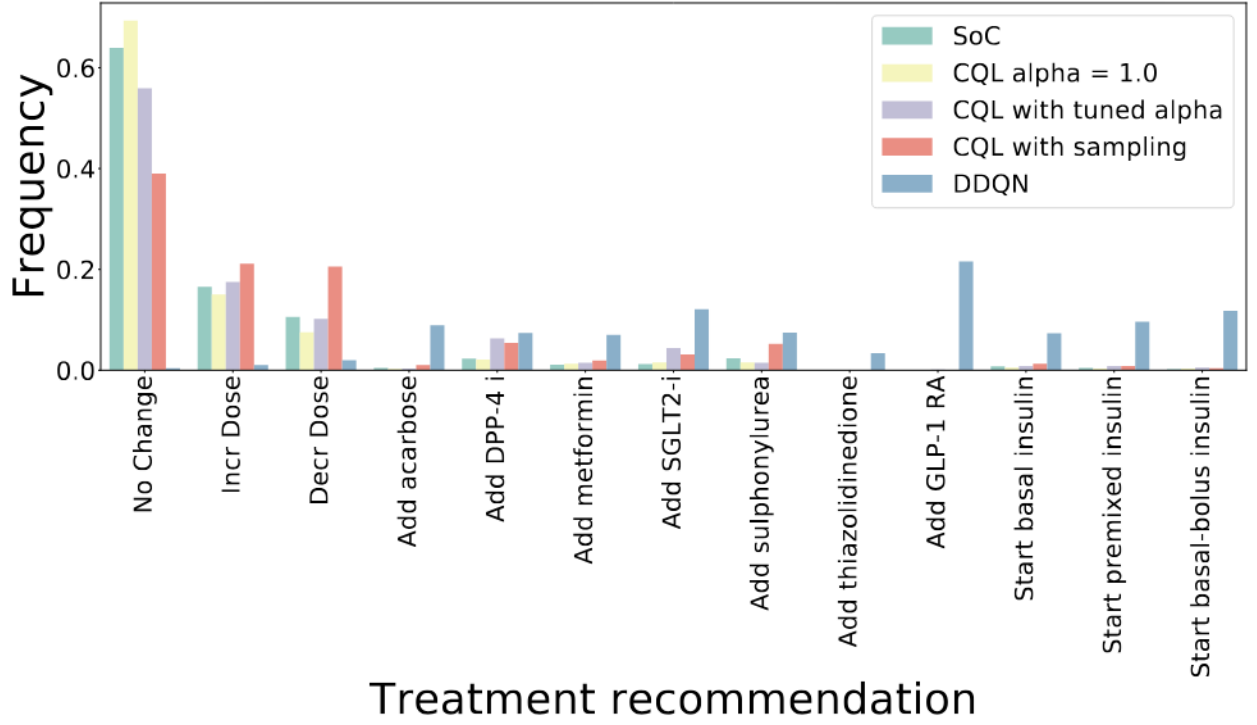


Figure 1: Distributions of unconstrained treatment recommendations under the various RL agents (as selected via grid-search) for the diabetes application.

with $\alpha = 1.0$, which has a WIS score of 3.653, the WIS scores are also higher for CQL with $\alpha = 0.1$ (5.530) and with $\alpha = 0.8$ (3.795); as well as for all the sampling schemes, with under+oversampling achieving the highest WIS score among all methods (5.721), followed by undersampling (5.082), then oversampling (4.070). Then, sampling and hyperparameter tuning both offer alternative means of improving the recommendations under CQL, with sampling outperforming hyperparameter tuning.

Sepsis treatment application. For the sepsis application as well, sampling and lowering α lead to greater divergence between the CQL and SoC recommendations. Fig. 2 shows that both reduce action imbalance, contributing to increases in the frequencies of the highest doses (top right section of each plot) of both IV fluids and vasopressors, as well as increases in the frequencies of higher IV fluid doses for the lowest vasopressor dose category. The difference is more pronounced for sampling than for hyperparameter tuning, and indeed the model concordance rate in Table 4 also decreases for all the sampling settings (from 30.1% for $\alpha = 1.0$ to between 25.3% and 28.5% for the different sampling settings), while no decrease is observed for any $\alpha, \alpha < 1$.

The WIS scores are also higher for CQL with $\alpha = 0.1$ (0.203), undersampling (0.191), oversampling (0.180), and under+oversampling (0.210), compared to CQL with $\alpha = 1.0$ (0.175). CQL with under+oversampling attains the highest score. From Equation 20, this implies that survival rates are expected to increase for these CQL agents over CQL with $\alpha = 1.0$. Thus, the finding that sampling and hyperparameter tuning can both improve the recommendations under CQL, with sampling outperforming hyperparameter tuning, generalizes from the diabetes to the sepsis application. The relative performance of the different sampling approaches also generalizes to the sepsis application. Under+oversampling outperforms undersampling, which in turn outperforms oversampling, in terms of WIS scores. This is in contrast to the finding in the supervised learning setting that oversampling tends to outperform undersampling [Zha et al., 2022]. A key difference between the supervised learning setting and our treatment optimization setting is that in the former, accuracy can suffer due to loss of information on majority class sample, while in the latter, it is more conservative and suboptimal actions that are overrepresented. Thus, undersampling is unlikely to be contributing to loss of information on the optimal action.

6.3 Evaluation of constrained recommendations

With constrained recommendations, the distribution across actions did not change noticeably from Fig. 1 (see Appendix C). Model concordance rates for the unconstrained and constrained recommendations of each CQL agent are

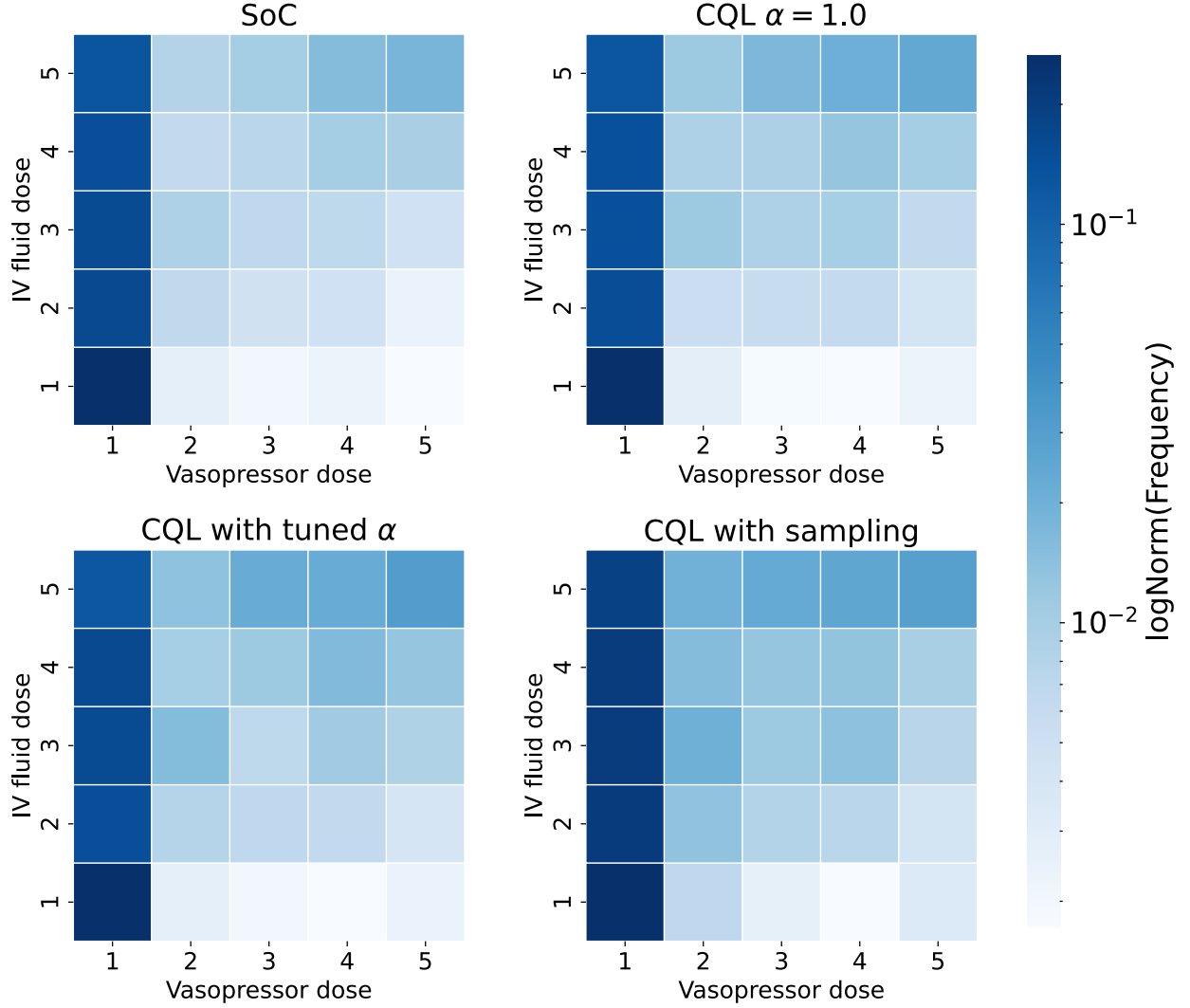


Figure 2: Distributions of treatment recommendations under SoC and the various RL agents (as selected via grid-search) on the sepsis dataset.

Table 4: Comparison of treatment recommendations by the SoC, and the unconstrained recommendations of the different RL agents, on the sepsis dataset. Metrics are Model Concordance Rate (MCR) and WIS score. The mean and standard deviations of the results across 3 seeds are reported. For each metric, the highest mean value across the RL agents is bolded, along with the associated standard deviation.

Agent	MCR	WIS score
SoC	100.0% \pm 0.0%	0.095 \pm 0.038
CQL with $\alpha = 1.0$	30.1% \pm 0.2%	0.175 \pm 0.018
CQL with $\alpha = 0.9$	30.3% \pm 0.2%	0.135 \pm 0.087
CQL with $\alpha = 0.8$	30.4% \pm 0.2%	0.147 \pm 0.056
CQL with $\alpha = 0.5$	31.3% \pm 0.2%	0.142 \pm 0.051
CQL with $\alpha = 0.1$	31.3% \pm 0.4%	0.203 \pm 0.075
CQL with undersampling	28.0% \pm 0.5%	0.191 \pm 0.053
CQL with oversampling	28.5% \pm 0.1%	0.180 \pm 0.044
CQL with under+oversampling	25.3% \pm 0.2%	0.210 \pm 0.035

largely similar (Table 2). This is due to the high constraint satisfaction rates for the unconstrained recommendations (Table 3). Consistent with the analysis in Section 4.3, the CQL agents’ WIS scores experience only a small drop.

Our other findings also generalize to the constrained recommendations. Compared to DDQN, CQL with $\alpha = 1.0$ exhibits higher model concordance with the SoC (62.6% vs. 1.6%), and achieves a higher WIS score (3.698 vs. -0.190), implying that CQL’s recommendations are more closely aligned with the SoC while also translating to improved expected health outcomes. The WIS scores are higher for CQL with $\alpha = 0.8$ (3.798), $\alpha = 0.1$ (5.496), undersampling (5.763), oversampling (4.049), and under+oversampling (4.668), compared to CQL with $\alpha = 1.0$ (3.698). CQL with undersampling attains the highest score. We conclude here as well that sampling and hyperparameter tuning can both improve over CQL recommendations, with sampling again outperforming hyperparameter tuning.

7 Conclusion

We have demonstrated that offline reinforcement learning (based on CQL) outperforms a popular deep off-policy RL method (DDQN) for a real-world diabetes treatment optimization application. We found that offline RL recommendations are not only more closely aligned to clinical practice, but also translate to substantial improvements in expected health outcomes. Further, to address the common challenges of action imbalance encountered in real-world treatment optimization tasks, we devised a practical but theoretically grounded offline RL strategy for transition sampling of training data. Via extensive experiments for two real-world treatment optimization applications, we demonstrated improvements with this strategy over off-policy (DDQN) and offline (CQL) RL baselines in terms of expected health outcomes, as well as in terms of alignment of the recommendations with clinical practice guidelines. Further, we showed theoretically and empirically that our results extend to when hard safety constraints are enforced via an intuitive heuristic. Our findings strongly suggest that offline RL should be chosen over off-policy RL for treatment optimization applications, as a means of enhancing safety and efficacy. Further, we highlight that transition sampling could find application in broader domains with critical safety considerations.

Future work could expand on our sampling approach by exploring more efficient automated sampling search strategies than grid-search. Further, while we made efforts to conduct principled and extensive evaluations across a holistic set of metrics, inherent limitations with off-policy evaluations based on retrospective data should be acknowledged. Hence, future work could also include qualitative evaluations by clinician experts, who could rate the acceptability of the RL agents’ recommendations.

Acknowledgements

This research is supported by A*STAR, Singapore under its Industry Alignment Pre-Positioning Fund (Grant No. H19/01/a0/023 – Diabetes Clinic of the Future). We thank Sing Yi Chia and Nur Nasyitah Mohamed Salim for data preparation assistance, and the SingHealth Duke-NUS Institute of Precision Medicine for provisioning the computational infrastructure for this work.

References

- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550 (7676):354–359, 2017.
- Aniruddh Raghu, Matthieu Komorowski, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach. In Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 147–163. PMLR, 18–19 Aug 2017a. URL <https://proceedings.mlr.press/v68/raghu17a.html>.
- Luca Roggeveen, Ali el Hassouni, Jonas Ahrendt, Tingjie Guo, Lucas Fleuren, Patrick Thorat, Armand RJ Girbes, Mark Hoogendoorn, and Paul WG Elbers. Transatlantic transferability of a new reinforcement learning model for optimizing haemodynamic treatment for critically ill patients with sepsis. *Artificial Intelligence in Medicine*, 112:102003, 2021. ISSN 0933-3657. doi:<https://doi.org/10.1016/j.artmed.2020.102003>. URL <https://www.sciencedirect.com/science/article/pii/S0933365720312689>.
- Xingzhi Sun, Yong Mong Bee, Shao Wei Lam, Zhuo Liu, Wei Zhao, Sing Yi Chia, Hanis Abdul Kadir, Jun Tian Wu, Boon Yew Ang, Nan Liu, Zuo Lei, Zhuoyang Xu, Tingting Zhao, Gang Hu, and Guotong Xie. Effective treatment recommendations for type 2 diabetes management using reinforcement learning: Treatment recommendation model

- development and validation. *J Med Internet Res*, 23(7):e27858, Jul 2021. ISSN 1438-8871. doi:10.2196/27858. URL <https://www.jmir.org/2021/7/e27858>.
- Hua Zheng, Ilya O. Rozhov, Wei Xie, and Judy Zhong. Personalized multimorbidity management for patients with type 2 diabetes using reinforcement learning of electronic health records. *Drugs*, 4:471–482, 2021. doi:<https://doi.org/10.1007/s40265-020-01435-4>.
- Huan-Hsin Tseng, Yi Luo, Sunan Cui, Jen-Tzung Chien, Randall K Ten Haken, and Issam El Naqa. Deep reinforcement learning for automated radiation adaptation in lung cancer. *Medical Physics*, 44:6690–6705, 2017. doi:10.1002/mp.12625.
- Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. *Reinforcement learning: State-of-the-art*, pages 45–73, 2012.
- Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Mar. 2016. doi:10.1609/aaai.v30i1.10295. URL <https://ojs.aaai.org/index.php/AAAI/article/view/10295>.
- Mingyu Lu, Zachary Shahn, Daby Sow, Finale Doshi-Velez, and Li-Wei H Lehman. Is deep reinforcement learning ready for practical applications in healthcare? a sensitivity analysis of duel-DDQN for hemodynamic management in sepsis patients. *AMIA Annu. Symp. Proc.*, 2020:773–782, 2020.
- Xuefeng Peng, Yi Ding, David Wihl, Omer Gottesman, Matthieu Komorowski, Li-Wei H Lehman, Andrew Ross, Aldo Faisal, and Finale Doshi-Velez. Improving sepsis treatment strategies by combining deep and kernel-based reinforcement learning. *AMIA Annu. Symp. Proc.*, 2018:887–896, December 2018.
- Chao Yu, Guoqi Ren, and Jiming Liu. Deep inverse reinforcement learning for sepsis treatment. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 1–3, 2019. doi:10.1109/ICHI.2019.8904645.
- Taiyu Zhu, Kezhi Li, Pau Herrero, and Pantelis Georgiou. Basal glucose control in type 1 diabetes using deep reinforcement learning: An in silico validation. *IEEE Journal of Biomedical and Health Informatics*, 25(4): 1223–1232, 2021. doi:10.1109/JBHI.2020.3014556.
- Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019a.
- Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.
- Justin Fu, Aviral Kumar, Matthew Soh, and Sergey Levine. Diagnosing bottlenecks in deep q-learning algorithms. In *International Conference on Machine Learning*, pages 2021–2030. PMLR, 2019.
- Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1179–1191. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf>.
- Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33: 14129–14142, 2020a.
- S Nemati, M.M. Ghassemi, and G. Clifford. Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2016a. doi:10.1109/EMBC.2016.7591355.
- Baiju R Shah, Janet E Hux, Andreas Laupacis, Bernard Zinman, and Carl van Walraven. Clinical inertia in response to inadequate glycemic control: do specialists differ from primary care physicians? *Diabetes Care*, 28(3):600–606, March 2005.
- Chao Yu, Jiming Liu, Shamim Nemati, and Guosheng Yin. Reinforcement learning in healthcare: a survey. *ACM Computing Surveys*, 33(1), 2021a.
- Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment, 2017b. URL <https://arxiv.org/abs/1711.09602>.
- Gabriel Schamberg, Marcus Badgeley, and Emery N. Brown. Controlling level of unconsciousness by titrating propofol with deep reinforcement learning. *AI in Medicine*, 2020.
- Chao Yu, Guoqi Ren, and Yinzhaodong. Supervised-actor-critic reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. *BMC medical informatics and decision making*, 2020b.

- Yiqin Yang, Xiaoteng Ma, Chenghao Li, Zewu Zheng, Qiyuan Zhang, Gao Huang, Jun Yang, and Qianchuan Zhao. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning, 2021. URL <https://arxiv.org/abs/2106.03400>.
- Qing Wang, Jiechao Xiong, Lei Han, peng sun, Han Liu, and Tong Zhang. Exponentially weighted imitation learning for batched historical data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL <https://proceedings.neurips.cc/paper/2018/file/4aec1b3435c52abddf8334ea0e7141e0-Paper.pdf>.
- Xinyue Chen, Zijian Zhou, Zheng Wang, Che Wang, Yanqiu Wu, and Keith Ross. Bail: Best-action imitation learning for batch deep reinforcement learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18353–18363. Curran Associates, Inc., 2020. URL <https://proceedings.neurips.cc/paper/2020/file/d55cbf210f175f4a37916eafe6c04f0d-Paper.pdf>.
- Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua M Susskind, Jian Zhang, Ruslan Salakhutdinov, and Hanlin Goh. Uncertainty weighted actor-critic for offline reinforcement learning. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11319–11328. PMLR, 18–24 Jul 2021. URL <https://proceedings.mlr.press/v139/wu21i.html>.
- Ilya Kostrikov, Jonathan Tompson, Rob Fergus, and Ofir Nachum. Offline reinforcement learning with fisher divergence critic regularization. In *International Conference on Machine Learning*, 2021.
- Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021b.
- Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2052–2062. PMLR, 09–15 Jun 2019. URL <https://proceedings.mlr.press/v97/fujimoto19a.html>.
- Aviral Kumar, Justin Fu, George Tucker, and Sergey Levine. *Stabilizing Off-Policy Q-Learning via Bootstrapping Error Reduction*. Curran Associates Inc., Red Hook, NY, USA, 2019b.
- Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning, 2019. URL <https://arxiv.org/abs/1911.11361>.
- Mehdi Fatemi, Mary Wu, Jeremy Petch, Walter Nelson, Stuart J. Connolly, Alexander Benz, Anthony Carnicelli, and Marzyeh Ghassemi. Semi-markov offline reinforcement learning for healthcare, 2022. URL <https://arxiv.org/abs/2203.09365>.
- Mehdi Fatemi, Taylor W Killian, Jayakumar Subramanian, and Marzyeh Ghassemi. Medical dead-ends and learning to identify high-risk states and treatments. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 4856–4870. Curran Associates, Inc., 2021. URL <https://proceedings.neurips.cc/paper/2021/file/26405399c51ad7b13b504e74eb7c696c-Paper.pdf>.
- Taylor W. Killian, Haoran Zhang, Jayakumar Subramanian, Mehdi Fatemi, and Marzyeh Ghassemi. An empirical study of representation learning for reinforcement learning in healthcare, 2020. URL <https://arxiv.org/abs/2011.11235>.
- Jiafei Lyu, Xiaoteng Ma, Xiu Li, and Zongqing Lu. Mildly conservative q-learning for offline reinforcement learning, 2022. URL <https://arxiv.org/abs/2206.04745>.
- Miroslav Kubat and Stan Matwin. Addressing the curse of imbalanced training sets: One sided selection. In *Proceedings of the 14th International Conference on Machine Learning*, pages 179–186. Morgan Kaufman, 1997.
- Charles X. Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD’98, page 73–79. AAAI Press, 1998.
- A.H. Schistad Solberg and R. Solberg. A large-scale evaluation of features for automatic detection of oil spills in ers sar images. In *IGARSS ’96. 1996 International Geoscience and Remote Sensing Symposium*, volume 3, pages 1484–1486 vol.3, 1996. doi:10.1109/IGARSS.1996.516705.
- Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, jun 2002. ISSN 1076-9757.

- Haibo He, Yang Bai, Eduardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008. doi:10.1109/IJCNN.2008.4633969.
- C.J.C.H. Watkins. *Learning from Delayed Rewards*. PhD thesis, University of Cambridge England, 1989.
- American Diabetes Association. Standards of Medical Care in Diabetes—2022 Abridged for Primary Care Providers. *Clinical Diabetes*, 40(1):10–38, 01 2022. ISSN 0891-8929. doi:10.2337/cd22-as01. URL <https://doi.org/10.2337/cd22-as01>.
- Daniel Yan Zheng Lim, Sing Yi Chia, Hanis Abdul Kadir, Nur Nasyitah Mohamed Salim, and Yong Mong Bee. Establishment of the singhealth diabetes registry. *Clinical Epidemiology*, 13:215–223, 2021. doi:<https://doi.org/10.2147/CLEP.S300663>.
- Matthieu Komorowski, Leo A. Celi, Omar Badawi, Anthony C. Gordon, and A. Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nat Med*, 24, Nov. 2018. doi:10.1038/s41591-018-0213-5. URL <https://www.nature.com/articles/s41591-018-0213-5>.
- Alistair E.W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. 3, May. 2016. doi:10.1038/sdata.2016.35. URL <https://www.nature.com/articles/sdata201635>.
- Tim Hesterberg. Weighted average importance sampling and defensive mixture distributions. *Technometrics*, 37(2): 185–194, 1995. ISSN 00401706. URL <http://www.jstor.org/stable/1269620>.
- Rongmei Lin, Matthew D. Stanley, Mohammad M. Ghassemi, and Shamim Nemati. A deep deterministic policy gradient approach to medication dosing and surveillance in the icu. In *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4927–4931, 2018. doi:10.1109/EMBC.2018.8513203.
- Shamim Nemati, Mohammad M. Ghassemi, and Gari D. Clifford. Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2978–2981, 2016b. doi:10.1109/EMBC.2016.7591355.
- Charles Teddlie and Fen Yu. Mixed methods sampling: A typology with examples. *Journal of Mixed Methods Research*, 1(1):77–100, 2007. doi:10.1177/1558689806292430. URL <https://doi.org/10.1177/1558689806292430>.
- Daochen Zha, Kwei-Herng Lai, Qiaoyu Tan, Sirui Ding, Na Zou, and Xia Hu. Towards automated imbalanced learning with deep hierarchical reinforcement learning, 2022.
- Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78:1–3, 1950.

A Theoretical analyses

A.1 Constraints: Proof of Property 1.

We can write the optimality gap of the RL model’s policy after applying the heuristic, $\pi_{RL,c}$, as follows:

$$\begin{aligned} & \mathbb{E}[V^{\pi_c^*}(s_0)] - \mathbb{E}[V^{\pi_{RL,c}}(s_0)] \\ = & \mathbb{E}[V^{\pi_c^*}(s_0)] - \mathbb{E}[V^{\pi^*}(s_0)] + \mathbb{E}[V^{\pi^*}(s_0)] - \mathbb{E}[V^{\pi_{RL}}(s_0)] + \mathbb{E}[V^{\pi_{RL}}(s_0)] - \mathbb{E}[V^{\pi_{RL,c}}(s_0)] \end{aligned} \quad (25)$$

$$\begin{aligned} \leq & \mathbb{E}[V^{\pi^*}(s_0)] - \mathbb{E}[V^{\pi_{RL}}(s_0)] \\ & + \mathbb{E}[V^{\pi_{RL}}(s_0) - V^{\pi_{RL,c}}(s_0) | \pi_{RL}(s_t) \in A_F(s_t) \forall t] \cdot \Pr[\pi_{RL}(s_t) \in A_F(s_t) \forall t] \\ & + \mathbb{E}[V^{\pi_{RL}}(s_0) - V^{\pi_{RL,c}}(s_0) | \exists t \text{ s.t. } \pi_{RL}(s_t) \notin A_F(s_t)] \cdot \Pr[\exists t \text{ s.t. } \pi_{RL}(s_t) \notin A_F(s_t)] \end{aligned} \quad (26)$$

$$\leq \mathbb{E}[V^{\pi^*}(s_0)] - \mathbb{E}[V^{\pi_{RL}}(s_0)] + \frac{\bar{r} - r}{1 - \gamma} \sum_{t=0}^T \Pr[\exists t \text{ s.t. } \pi_{RL}(s_t) \notin A_F(s_t)]. \quad (27)$$

The first inequality comes from observing that the first summand in Equation 25 is less than 0 by the optimality of π^* . The second inequality comes from the assumption that r is bounded, and applying the inequalities

$$\frac{r}{1 - \gamma} \leq \mathbb{E}[V^{\pi_{RL}}(s_0)] \leq \frac{\bar{r}}{1 - \gamma}.$$

This proves the theorem. \square

B Additional Implementation Details

B.1 Weighted Importance Sampling

To select a model for the clinician policy, we used the same seeds and train, test, and validation sets defined in Section 5.3; and trained multinomial logistic regression models using the `sklearn` library in Python. We set the one-hot-encoded SoC treatment recommendation as the dependent variable and the patient medical profile (state) as the independent variables. We tested different regularization strengths, $C \in \{0.1, 1, 10, 10^2, 10^3\}$, and balancing of class weights to mitigate action imbalance (10 hyperparameter settings). We selected the best model using the Brier score loss Brier [1950] on the validation set. Despite the action imbalance, models without balancing of class weights performed better. The results on the diabetes and sepsis test sets are reported in Table 5.

Table 5: Performance of the best clinician model on the diabetes and sepsis test sets, in terms of means and standard deviations across 3 seeds for the Brier score and AUC.

Metric	Diabetes	Sepsis
Brier score loss	0.487 ± 0.002	0.755 ± 0.003
AUC	0.830 ± 0.003	0.851 ± 0.002

B.2 Sampling

We applied stratified random sampling based on gender, ethnicity, age, and diabetes duration for the diabetes treatment setting, and based on gender, age, and whether the ICU stay was a readmission for the sepsis treatment setting. For continuous valued features, we binned values into 3 quantiles and stratified visits using the discretized values.

C Additional Results

Here, we supplement the results in Section 6. For the diabetes and sepsis settings, Tables 6 and 7 provide a sensitivity analysis of the performance of the performance of CQL with undersampling, oversampling, and under+oversampling for different K .

Finally, we complement Fig 1, which gives the distribution across treatment options of the RL models’ unconstrained recommended treatments, with Fig. 3, which shows this distribution for the constrained recommendations. This distribution is not noticeably different from the distribution of unconstrained actions given in Fig 1, i.e., the constrained actions exhibit the same trends. All the CQL models’ recommendations are more similar in distribution to the SoC than DDQN’s recommendations. Regularization parameter tuning and sampling lead to a greater divergence in distribution from the SoC recommendations compared to CQL with $\alpha = 1.0$.

Table 6: Comparison of the unconstrained and constrained treatment recommendations by CQL with undersampling, CQL with oversampling, and CQL with under+oversampling, on the diabetes dataset, for different K . Model Concordance Rate (MCR), WIS scores, and Appropriate Intensification Rates (AIR) are used as metrics. Results are the means and standard deviations across 3 seeds.

Sampling type	K	MCR		WIS score		AIR	
		Unconstrained	Constrained	Unconstrained	Constrained	Unconstrained	Constrained
Undersampling	0.4	$36.6\% \pm 0.8\%$	$37.4\% \pm 0.7\%$	5.082 ± 1.062	5.551 ± 1.191	$73.5\% \pm 3.2\%$	$72.2\% \pm 3.4\%$
Undersampling	0.8	$41.4\% \pm 6\%$	$42.0\% \pm 0.6\%$	5.738 ± 0.376	4.028 ± 2.176	$63.6\% \pm 0.5\%$	$64.6\% \pm 3.8\%$
Undersampling	1.2	$43.6\% \pm 0.2\%$	$44.1\% \pm 0.2\%$	5.297 ± 0.816	4.441 ± 1.675	$60.9\% \pm 0.3\%$	$62.7\% \pm 4.5\%$
Oversampling	0.4	$61.9\% \pm 2.6\%$	$62.0\% \pm 2.5\%$	5.037 ± 1.112	5.064 ± 1.177	$36.8\% \pm 1.7\%$	$36.3\% \pm 1.6\%$
Oversampling	0.8	$61.5\% \pm 0.5\%$	$61.7\% \pm 0.4\%$	4.070 ± 1.374	4.049 ± 1.342	$40.8\% \pm 1.6\%$	$40.3\% \pm 1.7\%$
Under+oversampling	1.0	$41.3\% \pm 0.5\%$	$42.0\% \pm 0.5\%$	5.721 ± 0.392	4.668 ± 1.692	$76.1\% \pm 10.1\%$	$68.5\% \pm 8.9\%$

Table 7: Comparison of the unconstrained and constrained treatment recommendations by CQL with undersampling, CQL with oversampling, and CQL with under+oversampling, on the diabetes dataset, for different K . Model Concordance Rate (MCR) and WIS scores are used as metrics. Results are the means and standard deviations across 3 seeds.

Sampling type	K	MCR	WIS score
Undersampling	0.4	25.9% \pm 0.2%	0.122 \pm 0.102
Undersampling	0.8	27.9% \pm 0.3%	0.142 \pm 0.112
Undersampling	1.2	28.0% \pm 0.5%	0.248 \pm 0.082
Oversampling	0.4	28.5% \pm 0.3%	0.210 \pm 0.014
Oversampling	0.8	28.5% \pm 0.1%	0.174 \pm 0.037
Under+oversampling	1.0	25.3% \pm 0.2%	0.210 \pm 0.035

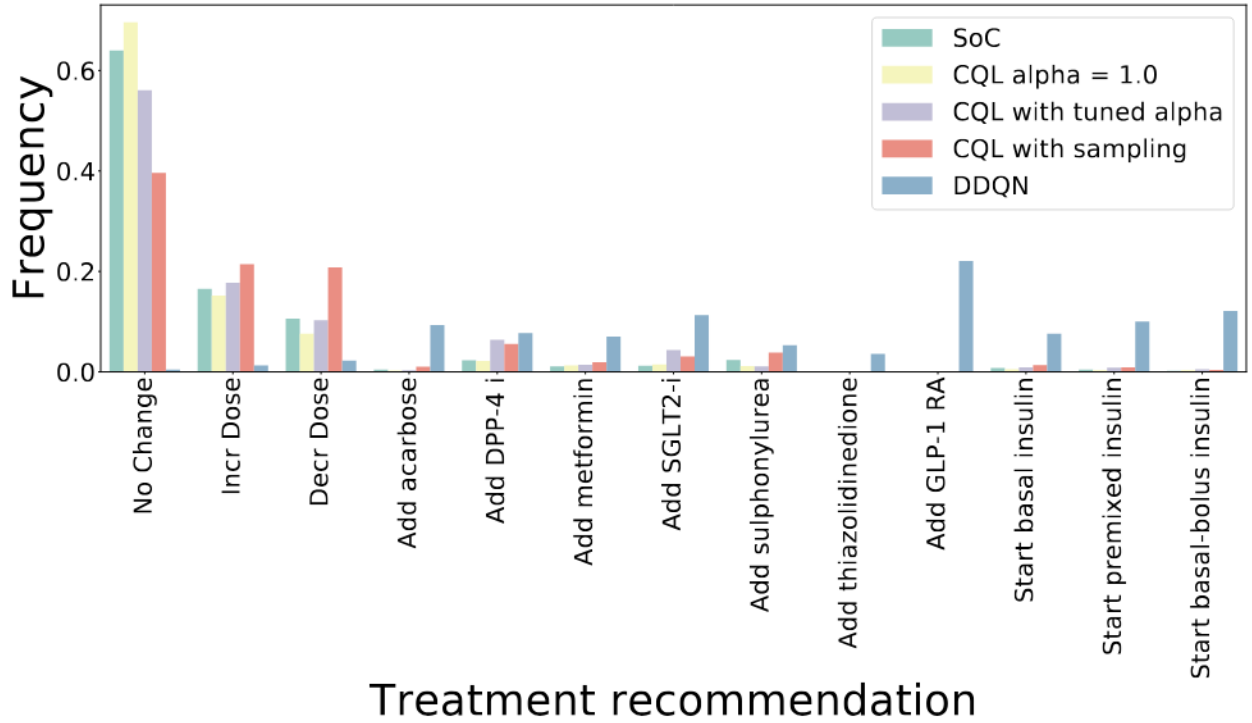


Figure 3: Distribution of the SoC treatment recommendations and the constrained treatment recommendations under CQL and DDQN for the diabetes setting. The RL agents are selected via grid-search across all hyperparameter combinations, where the agent with the highest WIS score for constrained recommendations on the validation set is selected.