Semi-Markov Offline Reinforcement Learning for Healthcare

Mehdi Fatemi[*] mehdi.fatemi@microsoft.com
Microsoft Research

Mary Wu[*] mary@cs.toronto.edu
University of Toronto

Jeremy Petch petchj@hhsc.ca
Hamilton Health Sciences

Walter Nelson nelsonwa@hhsc.ca
Hamilton Health Sciences

Stuart J. Connolly stuart.connolly@phri.ca
Population Health Research Institute

Alexander Benz alexander.benz@phri.ca
Population Health Research Institute

Anthony Carnicelli anthony.carnicelli@duke.edu
Duke University

Marzyeh Ghassemi mghassem@mit.edu
Massachusetts Institute of Technology

Abstract

Reinforcement learning (RL) tasks are typically framed as Markov Decision Processes (MDPs), assuming that decisions are made at fixed time intervals. However, many applications of great importance, including healthcare, do not satisfy this assumption, yet they are commonly modelled as MDPs after an artificial reshaping of the data. In addition, most healthcare (and similar) problems are *offline* by nature, allowing for only retrospective studies. To address both challenges, we begin by discussing the Semi-MDP (SMDP) framework, which formally handles actions of variable timings. We next present a formal way to apply SMDP modifications to nearly any given value-based offline RL method. We use this theory to introduce three SMDP-based offline RL algorithms, namely, SDQN, SDDQN, and SBCQ. We then experimentally demonstrate that only these SMDP-based algorithms learn the optimal policy in variable-time environments, whereas their MDP counterparts do not. Finally, we apply our new algorithms to a real-world offline dataset pertaining to *warfarin dosing for stroke prevention* and demonstrate similar results.

Data and Code Availability Access to the Randomized Control Trial data used for the warfarin study is governed by the COMBINE AF executive committee. Investigators interested in working with these data should contact a member of the committee to discuss potential collaboration (Carnicelli et al., 2021). The code is available at https://github.com/mary-wu/smdp.

1. Introduction

Many healthcare and medical problems are sequential decision-making tasks: the patient's health condition is monitored, medical interventions are administered accordingly, and the cycle repeats until a termination point. A formal way to model such problems is through Markov Decision Processes (MDPs) (Puterman, 1994). In the MDP setting, patients are often modelled by their health *state*. The state accounts for a sufficient history of the patient's health records, as well as demographic or other relevant information, which together make the state self-contained at any time point (Ghassemi et al., 2014, 2015; Wu et al., 2016; Ghassemi et al., 2017; Suresh et al., 2017; Raghu et al., 2017). Solving an MDP may be achieved through a wealth of algorithmic approaches,

[*] These authors contributed equally

which are collectively known as *reinforcement learning* (RL) (Bertsekas and Tsitsiklis, 1996; Sutton and Barto, 2018). However, there are two significant challenges in many healthcare problems. First, observations are sporadic, which violates the MDP's required regular spacing of time intervals. Second, datasets are typically offline without an opportunity to interact. This paper will propose a class of algorithms to address both concerns.

A motivating example is warfarin dose management. Warfarin is an anticoagulant commonly prescribed to reduce the risk of stroke associated with atrial fibrillation (Pirmohamed, 2018). This example is of particular interest as stroke is the second leading cause of death worldwide (WHO), warfarin continues to be used around the globe, and clinicians struggle to prescribe it effectively because of its complex pharmacodynamics. Warfarin dosing for atrial fibrillation is easily modelled as a sequential decision-making task: patients visit the clinic at certain time intervals to monitor their health, during which a physician may adjust their dose. At each visit, the patient's International Normalized Ratio (INR) is measured and used, along with the patient's medical history and demographic information, to potentially modify the dosage. Although this is a well-defined decision-making process, we cannot directly model this as an MDP due to inconsistent times between clinical visits (see Section 6.2).

In general terms, the MDP framework contains no notion of a course of action persisting over a variable period of time (Sutton et al., 1999). More precisely, the standard MDP framework does not include temporal abstraction or temporally extended actions. Temporal abstraction can be introduced into RL in a variety of ways, see for example Ring (1991); Wixson (1991); Schmidhuber (1991); Singh (1992); Chrisman (1994); Sutton (1995); Precup et al. (1997); Precup and Sutton (1997). Among these methods, using *options* (Precup et al., 1998) has versatility, solid mathematical exposition, and similarity of the resultant theory to the standard RL framework. An option is a fixed policy with an initiation function, which signals where the policy can be selected, and a termination function, which specifies when the policy is terminated while it is running. While an option is active, the primitive actions are chosen according to its policy until the termination function becomes true, at which point the next option is chosen. Options have been utilized to formally introduce temporal abstractions to the MDP framework (Sutton et al., 1999) through the Semi-MDP (SMDP) framework (Bradtke and Duff, 1994; Parr, 1998), by replacing actions with options. This makes SMDPs appropriate for modeling unregulated discrete-event problems over continuous or discrete time. There are several other methods that are built on continuous-time dynamic programming (CDP). For example, Marked Temporal Processes (Aalen et al., 2008; Zarezade et al., 2017) are a domain-specific application of CDP (they use classic HJB results, which are also core to SMDP theory (Bradtke and Duff, 1994)). We however note that none of these methods are considered inherently different methodologies; hence, we do not discuss them in this paper nor compare our results against them. Importantly, we also remark that none of these classic or recent materials pertain to the offline case, which we discuss next.

Another important characteristic of many healthcare and treatment management problems is their *offline* nature (Lange et al., 2012), i.e., the dataset is static and must be analysed in a retrospective manner due to diverse safety, ethical, or legal constraints (Fatemi et al., 2021). While offline RL literature has advanced substantially in recent years, proposed offline RL algorithms still target MDP scenarios (Lange et al., 2012; Fujimoto et al., 2019b). One core contribution of this paper is to introduce a systematic way to construct SMDP versions of common offline RL methods. This new class of algorithms is sound and easy to implement once the MDP version is given. Of particular interest, we introduce the SBCQ algorithm, an SMDP version of Batch-Constrained Q-learning (BCQ) (Fujimoto et al., 2019b). SBCQ addresses the over-estimation issue of offline RL methods the same way that BCQ does, but extends the framework to handle Semi-Markov regimes with arbitrary options.

While options are great tools to properly address the irregular timing of healthcare scenarios, the standard SMDP framework requires full observability of intra-option rewards. This is not normally the case in healthcare (and perhaps various other similar domains), since both the state and the reward are monitored solely at the time that the patient comes to the clinic. To address this issue, we choose to use linear interpolation and will argue how this solution is medically sound in the case of the stroke problem. In addition, we are interested in the case where options are given rather than learned. This is because in our healthcare setting (and similar problems), the options are used to model the maintenance of a given prescription for various time intervals.

The rest of the paper is organized as follows: Section 2 discusses important related work. Section 3 explains the setting and reviews important results from the literature, which sets the stage for our main technical contributions, presented in Section 4. We then present a simple domain in Section 5 to concretely demonstrate how our methods work in comparison to the MDP variations, and contrast SBCQ with simpler SMDP algorithms. Finally, in Section 6, we apply our methods to the warfarin dosage management problem and report similar conclusions in real-world settings.

2. Related Work

RL for Health: RL has been the subject of much focus in health (Yu et al., 2019), with particular emphasis on sepsis with various goals: seeking to develop optimal treatment recommendation policies (Henry et al., 2015; Futoma et al., 2017; Komorowski et al., 2018; Saria, 2018; Raghu et al., 2017; Peng et al., 2018; Li et al., 2019; Tang et al., 2020b), treatment avoidance (Fatemi et al., 2021), and learning set-valued policies with the goal of human-in-the-loop treatment selection (Tang et al., 2020a). All such prior work aggregate observations at equally distributed time points such that these processes can be modelled as MDPs. These are in direct contrast to our approach that allows arbitrary time stamps for observations.

SMDP in Deep RL: While SMDP theory has a long history, its deployment to Deep RL settings is fairly recent and limited. For example, Schmoll and Schubert (2020) uses SMDP in a DQN setting for the task of collecting stochastic, spatially distributed resources (stochastic resource collection), and Bellinger et al. (2020) broadly applies SMDP to DQN and Recurrent DQN. However, to the best of our knowledge, none of the few existing publications investigate offline scenarios, which is one of the core considerations of the current paper.

Offline RL: Offline RL has been studied for a long time in the context of batch RL (Lange et al., 2012). Importantly, the performance of *off-policy* RL algorithms (Sutton and Barto, 2018) degrade drastically in fully *offline* or batch settings as they need to sufficiently explore the environment with additional interactions (Jaques et al., 2019; Fujimoto et al., 2019b). The challenges are significantly intensified when the dataset is limited and exploration of various states and actions is not available (Bertsekas and Tsitsiklis, 1996; Kushner and Yin, 2003). Overfitting to data-collection artifacts in offline cases is another significant issue (François-Lavet et al., 2019; Sinha and Garg, 2021; Agarwal et al., 2020). Moreover, estimation errors due to limited data may further lead to mistimed or inappropriate decisions with adverse safety consequences (Rebba et al., 2006). For a more recent, yet partial, survey of MDP-based offline RL methods see Levine et al. (2020). Since most healthcare and medical-related sequential decision-making/reasoning problems are offline, this paper studies the proposed SMDP algorithms in fully offline scenarios.

3. Background

3.1. Markov Decision Processes

We consider discrete-time episodic Markov Decision Processes (MDPs), where an agent interacts with the environment at each of the lowest-level discrete time steps $t = 1, 2, \dots$. An MDP is defined as a tuple $(\mathcal{X}, \mathcal{A}, R, T, \gamma)$. $\mathcal{X}$ and $\mathcal{A}$ are respectively the state and action spaces, T is the transition kernel $T(x, a, \cdot) = P(\cdot|x, a)$ that gets the probability of transitions if action a is selected at state x, $\gamma \in [0, 1]$ is the discount factor, and $R : \mathcal{X} \times \mathcal{A} \times \mathcal{X} \mapsto \mathbb{R}$ is the reward function. A stationary policy π maps each state $x \in \mathcal{X}$ to a probability distribution over the action space $\mathcal{A}$. We define a terminal state as the final state at which the environment terminates (in healthcare, it corresponds to the last point of a patient's recorded trajectory). The set of all terminal states is denoted by $\mathcal{X}_T \subset \mathcal{X}$. Mathematically, a terminal state is absorbing (self-transition w.p. 1) with zero reward afterwards. All terminal states are zero-valued, but the transitions to them may be associated with a non-zero reward.

On each time step t, the agent perceives the environment's state, $X_t = x$, and selects a primitive action, $A_t = a$, according to a policy $\pi(\cdot|x)$. The action influences the state and together with the internal dynamics of the environment, it transitions to a new state $X_{t+1} = x'$ at the next discrete time $t + 1$. This transition may incur a reward of $R_{t+1}(x, a, x')$. We define *return* as the sum of discounted rewards as the agent interacts with the environment. The value function corresponding to a policy π is denoted by $Q^\pi(x, a)$ and defined as the expected return if the agent selects action $A_t = a$ at state $X_t = x$ and acts according to π afterwards.

We define optimal value and state-value functions, respectively as $Q^*(x,a) = \max_\pi Q^\pi(x,a)$ and $V^*(x) = \max_\pi V^\pi(x) = \max_{a'} Q^*(x,a')$.

3.2. Options and Semi Markov Decision Processes

Semi-Markov Decision Processes (SMDPs) are a generalization of MDPs that handle actions with variable duration. More precisely, an SMDP is similar to an ordinary MDP, with the difference being that transitions may have stochastic time duration (Parr, 1998). The original SMDP theory treats the extended actions as black boxes over continuous time (Bradtke and Duff, 1994; Parr, 1998). To have a more versatile view, Sutton et al. (1999) built on that and introduced a superimposed version, where each extended action is seen as an *option*. We adopt their version in this paper and review the formalism in the rest of this section.

An option is defined as the tuple $(\mathcal{I}, \pi, \beta)$, where $\pi : \mathcal{X} \times \mathcal{A} \mapsto [0,1]$ is a policy, $\mathcal{I} \subseteq \mathcal{X}$ is the initiation set where the policy can be chosen, and $\beta : \mathcal{X} \mapsto [0,1]$ is the termination function. If an option is active, actions are selected according to its policy π until the option terminates stochastically according to β. For instance, the option of "opening-the-door" specifies the (possibly stochastic) sequence of all the primitive actions (e.g., all the muscle contractions) involved in reaching the door knob, turning it, pulling/pushing the door with proper force, and releasing the knob. More precisely, the sequence of such primitive actions can be seen as initiating a policy π at some proper set of states $\mathcal{I}$ (e.g., where the knob is reachable), running, and then terminating it once the knob is released (where $\beta = 1$).

Policies and termination functions are called Markov if they are only a function of the current state. On the other hand, an option's policy is called *Semi-Markov* if it is a function of any event from the *history* of the option. If an option is selected at time t, we define the history from t to any time $t' > t$ by $h_{tt'} = \{s_t, a_t, r_{t+1}, s_{t+1}, \ldots, s_{t'}\}$. Remark that Semi-Markov policies are less unconstrained than non-stationary policies in that Semi-Markovness only allows being function of events back to time t, but not before that. Similarly, we can define the termination function β to be Semi-Markov if it is a function of the option's history. For an option $o = (\mathcal{I}, \pi, \beta)$, we say o is Markov if both π and β are Markov, and we say o is Semi-Markov if any of π or β is Semi-Markov.

The concept of policy can be applied to options as well. Formally, if $\mathcal{O}$ is a set of options defined on an MDP $M = (\mathcal{X}, \mathcal{A}, R, T, \gamma)$, a policy $\mu(o|x)$ is defined as the probability of selecting $o = (\mathcal{I}, \pi, \beta) \in \mathcal{O}$ at state $x \in \mathcal{X}$. If $x \notin \mathcal{I}$, then $\mu(o|x) = 0$. Hence, a policy over a set of options induces a *flat policy* at the primitive actions level by selecting an option, waiting until it terminates, selecting the next option and so on. The flat policy is defined as the concatenation of all the option-level policies. Note that the flat policy is unlikely to be Markov even if both μ and all the options are Markov. We next extend the concept of value functions to *Semi-Markov flat policies* as

$$V^\pi(x) \doteq \mathbb{E}\left[\sum_{j=0}^{\infty} \gamma^j R_{t+j+1} | \mathcal{E}(\pi, x, t)\right]$$

where $\mathcal{E}(\pi, x, t)$ denotes the event of π being initiated at state x at time t. By definition, the value of a state under an option-level policy can then be defined as the value of the state under the corresponding flat policy: $V^\mu(x) \doteq V^{\mathsf{flat}(\mu)}(x)$ for all $x \in \mathcal{X}$. We also define $Q^\mu(x,o)$ as the value of selecting option $o = (\mathcal{I}, \pi, \beta) \in \mathcal{O}$ at state $x \in \mathcal{I}$ under policy μ; formally,

$$Q^\mu(x,o) \doteq \mathbb{E}\left[\sum_{j=0}^{\infty} \gamma^j R_{t+j+1} | \mathcal{E}(o\mu, x, t)\right]$$

where $o\mu$ denotes the Semi-Markov policy (composing o and μ) that first follows the policy of o until it terminates and then starts choosing another option according to μ in the resultant state and follows its policy, and so on.

SMDPs (Bradtke and Duff, 1994) are defined in a similar way as the MDPs, but with temporally extended actions, the selection of which depends on the history from each given state. Theorem 1 of (Sutton et al., 1999) states that any fixed set of options defined on an MDP establishes a discrete time SMDP. That is, the base problem is an MDP, over which extended actions are shaped. This view provides flexibility in terms of what happens inside each option, which can be examined, altered, learned, and planned.

We are interested in the case where a fixed set of options is given and we would like to determine the best strategy for selecting among the options when the current option terminates. Let us denote such a set of options as $\mathcal{O}$. Remark that if $\mathcal{O}$ does not contain all the primitive actions and/or any of the options lasts for more than one step, then achieving the optimal values at the level of primitive actions may *not* be feasible. Still, it is important to achieve the best results given the options. Of note, having more flexibility in terms of option design can improve the ultimate performance. For example, it can be proved that having the choice to terminate the current option whenever the agent wants (called option interruption) directly improves the values of all states. This extension can easily be applied using our pipeline. However, at this stage we assume that no observation is available between subsequent visits; hence, option interruption is also infeasible.

Let $\mu : \mathcal{X} \times \mathcal{O} \mapsto [0,1]$ be an option-level policy that selects options from $\mathcal{O}$ at states where the options' initiation sets allow, and let $\Pi(\mathcal{O})$ denote the set of all such policies. Expanding the optimal value function at the option level, it then follows:

$$\begin{aligned} V^*_{\mathcal{O}}(x) &\doteq \max_{\mu \in \Pi(\mathcal{O})} V^{\mu} \\ &= \max_{o \in \mathcal{O}} \mathbb{E}[R_{t+1} + \cdots + \gamma^{k-1} R_{t+k} \\ &\qquad + \gamma^k V^*_{\mathcal{O}}(x_{t+k}) \mid \mathcal{E}(o,x,t)] \\ &= \max_{o \in \mathcal{O}} \mathbb{E}[\rho(x,o) + \gamma^k V^*_{\mathcal{O}}(x') \mid \mathcal{E}(o,x)] \end{aligned} \tag{1}$$

where k is the duration of o when taken at x, x' is the state where the option terminates, and $\rho(s,o)$ is the *discounted cumulative rewards* in the course of option's operation. We also dropped t for clarity. Similarly, $Q^*_{\mathcal{O}}$ satisfies:

$$Q^*_{\mathcal{O}}(x,o) = \mathbb{E}[\rho(x,o) + \gamma^k \max_{o' \in \mathcal{O}} Q^*_{\mathcal{O}}(x',o') \mid \mathcal{E}(o,x)] \tag{2}$$

Importantly, remark that these Bellman equations incur a factor of γ^k for the bootstrapping term. This is to be expected since x' is k steps away from x. Nevertheless, this multiplier, along with the fact that $\rho(x,o)$ is the discounted return rather than the immediate reward, directly impact learning the optimal values. Equations (1) and (2) can directly be used in planning algorithms such as (SMDP-) value- or policy-iteration (see Sutton et al. (1999) for details and examples). Similarly, for TD-like learning algorithms, the update rule must be modified accordingly. Most prominently, the *SMDP Q-learning* update (Bradtke and Duff, 1994; Sutton et al., 1999) takes the following form:

$$\begin{aligned} Q(x,o) \longleftarrow\ & (1-\alpha) Q(x,o) \\ & + \alpha \left[\rho(x,o) + \gamma^k \max_{o' \in \mathcal{O}} Q(x',o') \right] \end{aligned} \tag{3}$$

where α is the learning rate. The SQ-learning algorithm of (3) is guaranteed to converge to the $Q^*_{\mathcal{O}}$ under similar conditions to the standard Q-learning (Watkins, 1989; Watkins and Dayan, 1992).

The core technical contribution of our work is to apply similar modifications to the update rules of state-of-the-art offline RL algorithms to yield their SMDP counterparts.

4. SMDP-based Offline RL

We introduce offline SMDP methods: a new class of algorithms that operate for variable-timed SMDP settings in an offline manner.

4.1. Algorithm Design

We target offline *value-based* algorithms in this paper. The core technique is as follows: we first update the Bellman target for (value-based) deep RL methods along the same lines as the SMDP Q-learning update rule. Next, we add algorithm-specific tricks to achieve the SMDP version of the algorithm. Specifically, in the MDP scenarios, the Q function is typically approximated using a parameterised function Q_θ, where θ is a vector of parameters. At each iteration j, the parameter vector is trained by minimising a sequence of squared temporal-difference errors:

$$\begin{aligned} L_j(\theta_j) &\doteq \mathbb{E}_{(x,a,r,x') \sim U(\mathcal{D})}\ \delta_j^2 \\ \delta_j &\doteq r + \gamma \max_{a'} Q_{\theta_j}(x',a') - Q_{\theta_j}(x,a) \end{aligned} \tag{4}$$

where $U(\mathcal{D})$ denotes uniform sampling from the offline data $\mathcal{D}$, and δ_j is the TD error at iteration j.

We first construct a new data set $\mathcal{D}'$ at the option level in the form of $\{(x,o,\rho,x',k)\}$, where x and x' are the states where o is initiated and terminated, respectively; ρ is the *discounted accumulated rewards* in the course of o; and k is the length of o. We then replace (4) with the option error as the following:

$$\begin{aligned} L_j(\theta_j) &\doteq \mathbb{E}_{(x,o,\rho,x',k) \sim U(\mathcal{D}')}\ \delta_j^2 \\ \delta_j &\doteq \rho + \gamma^k \max_{o'} Q_{\theta_j}(x',o') - Q_{\theta_j}(x,o) \end{aligned} \tag{5}$$

Algorithm 1: Generic algorithm for SMDP-based offline RL.

```
Data: $\mathcal{D}'$
Input: randomly initialized $Q_\theta$, $\gamma$
Output: learned $Q_\theta$
$j \longleftarrow 0$   /* $j :=$ total iteration counter */
for $epoch = 1, M$ do
    $\widetilde{\mathcal{D}} \longleftarrow \mathcal{D}'$
    repeat
        sample minibatch $\{(x_i, o_i, \rho_i, x'_i, k_i)\} \sim U(\widetilde{\mathcal{D}})$   /* $i :=$ index inside the minibatch */
        for all $i$ do
            $y_i \longleftarrow \begin{cases} \rho_i & \text{if } x'_i \text{ is terminal} \\ \rho_i + \gamma^{k_i} \max_{o'_i} Q_{\theta_j}(x'_i, o'_i) & \text{otherwise} \end{cases}$
            $\delta_{j,i} \longleftarrow y_i - Q_{\theta_j}(x_i, o_i)$
        end
        perform a gradient descent step on $\sum_i \delta^2_{j,i}$ with respect to the network parameters $\theta_j$
        $j \longleftarrow j + 1$
        $\widetilde{\mathcal{D}} \longleftarrow \widetilde{\mathcal{D}} \setminus \{(x_i, o_i, \rho_i, x'_i, k_i)\}$
    until $\widetilde{\mathcal{D}}$ is empty
end
```

Algorithm 1 presents a vanilla SMDP offline RL algorithm based on (5). Various more sophisticated algorithms can follow (5) immediately; we present two basic ones in this section and a more advanced one in the next section. In all such algorithms, the relevant lines of Algorithm 1 should properly be modified. Of note, to mitigate the *value overflow* problem (Fatemi et al., 2019), in all of our methods we also clip the $\max_{o'} Q_{\theta_j}(x', o')$ part of (5) to remain between the minimum and maximum possible returns, when applicable.

SDQN. Using the loss function from (5), we can then generalize the original Deep Q-networks (DQN) method (Mnih et al., 2015): We introduce *SDQN* by considering a main network Q_θ and a target network $Q_{\theta'}$, which is updated from the main network every K steps. The loss function (5) is then modified by replacing $\max_{o'} Q_{\theta_j}(x', o')$ with $\max_{o'} Q_{\theta'_j}(x', o')$.

SDDQN. In a similar way, we can extend the double DQN (DDQN) algorithm (Hasselt et al., 2016), which suppresses over-estimation issues of DQN by decoupling action-selection from maximization in the TD error. Our modification yields the new *SDDQN* variation with the following TD error:

$$\delta_j \doteq \rho + \gamma^k Q_{\theta'_j}\left(x', \text{argmax}_{o'} Q_{\theta_j}(x', o')\right) - Q_{\theta_j}(x, o)$$

4.2. SMDP Batch-Constrained Q-Learning

One of the challenges of offline RL methods is mitigating the adverse effects of a phenomenon called extrapolation error (Fujimoto et al., 2019b). There are a few causes of extrapolation error, such as insufficient and/or imbalanced data, model bias, and training mismatch. In all these cases, off-policy learning methods may yield an arbitrarily inaccurate value estimate $Q(s, a)$ and a sub-optimal policy. To overcome this, Batch-Constrained Q-learning (BCQ) restrains the action space during the training process to actions that are likely to be observed with a probability larger than threshold τ (Fujimoto et al., 2019b).

Central to BCQ is three core steps: (1) *minimize* the distance of selected actions to the data in the batch, (2) find states where observed data is similar to the batch, and (3) *maximize* the value function based on those. In the continuous setting, BCQ trains a generative model $G_\omega(a_i|x)$, commonly a conditional variational auto-encoder (Kingma and Welling, 2013; Sohn et al., 2015), to estimate the state-conditioned marginal likelihood of observing action a_i at state s. Actions are sampled from this network, and the highest-valued action is selected. In the discrete case, which is used in this paper, the same core principles are maintained with a simpler approach (Fujimoto et al., 2019a). $G_\omega(a_i|x)$ is instead a behavioural cloning network trained with a cross-entropy loss and used to mask actions whose relative probability are below a threshold τ (i.e. unlikely to be seen at the given state).

Originally, BCQ maintains four neural networks: two main networks $Q_{\theta_1}, Q_{\theta_2}$, and two target networks $Q_{\theta'_1}$, $Q_{\theta'_2}$. To compute the loss for the training of both main networks, the target value for a transition (x, a, x') is given by the following:

$$y = r(x, a, x') + \gamma \max_{a_i}[\lambda \min_{j=1,2} Q_{\theta'_j}(x', a_i) + (1-\lambda) \max_{j=1,2} Q_{\theta'_j}(x', a_i)] \quad (6)$$

where a_i's are such that $\frac{G_\omega(a_i|x)}{\max_{\hat{a}} G_\omega(\hat{a}'|x)} > \tau$ for some τ, and $\lambda \in (0,1)$. In the discrete case, BCQ simply uses a double-DQN form as discussed in the previous section. The target networks are updated from the main networks either as a lump update once every K steps, or gradually by $\theta'_j = \kappa\theta_j + (1-\kappa)\theta'_j$ with some fixed $\kappa \in (0,1)$.

The resulting policy is then as follows:

$$\pi(x) = \text{argmax}_{a'} Q_\theta(x, a'),\ a' \text{ s.t. } \frac{G_\omega(a'|x)}{\max_{\hat{a}} G_\omega(\hat{a}'|x)} > \tau$$

In order to adopt BCQ and construct its SMDP version, which we term *SBCQ*, we need to modify (6) in accordance with (5). Remark that in (6), the entire part in brackets yields Q of the next state, x', on which the max operator applies under the batch-constraint. We therefore need to replace γ with γ^k, r with ρ, and replace actions with options. Additionally, the generative model, $G_\omega(a_i|x)$, should be replaced with an option-level version, $G_\omega(o_i|x)$, and the action generation process should be performed over options instead. The rest of the algorithmic procedures in the original BCQ remains the same.

4.3. Estimation Requirements

In the algorithms presented above, the constructed dataset, D', contains transitions in the form of (x, o, ρ, x'). Hence, only the first and last states where o begins and ends are required to be known, not any others during the execution of o. However, the rewards at all the *intra* transitions are needed for computing the discounted return ρ. On the other hand, if the environment is SMDP but modeled as an MDP with aggregation of data at primitive time steps, then we will also require estimating states and actions in addition to rewards at all the primitive steps. This is in general significantly more limiting and is normally error prone, as we further will see in Section 6.2. It is also worth noting that in some problems, domain knowledge may help determine how to compute ρ (even fully accurately), which is rarely the case for states and actions. For example, in goal-oriented tasks, the reward is always zero except when the goal is achieved, where the last option also terminates. Hence, $\rho = 0$ for all options except the last one, where $\rho = \gamma^{k-1} r_g$, with r_g being the goal reward.

To estimate the rewards in the warfarin problem in particular, we refer to the medical literature and assume linear changes in subsequent INR measurements, as suggested by Rosendaal et al. (1993). This is because the goal of warfarin dosing management is to maximize the time we spend in the INR therapeutic range of 2-3, also known as the time in therapeutic range (TTR). To encourage this behaviour, the reward signal is +1 if the subsequent INR is within the therapeutic range and 0 otherwise. To obtain intra transition rewards, we can use linear interpolation to estimate the INR between clinical visits, and use these estimates to determine the reward signals and thus the discounted return ρ.

5. Illustrative Example

To demonstrate the need for SMDP-based algorithms in time-variable environments, we compare the MDP and SMDP variations for each of the three algorithms presented in Section 4 in a simple Minigrid environment (Chevalier-Boisvert et al., 2018). We intentionally design the environment such that only options are accessible to the agent and observations are only provided when options are terminated. Experimental details can be found in the Appendix.

5.1. Environment Setup

In the Minigrid, shown in Figure 1, the agent (red triangle) attempts to reach the goal state (green square) to achieve a reward of +10. The *primitive* actions in the environment are: turn left, turn right, move forward, and do nothing. Each option begins by selecting one of the primitive actions, followed by a series of forward steps. The number of steps depends on the agent's location in the grid — when the agent is in the leftmost column facing downwards, the step size is 4. When the agent is in the top row, the step size is 1. Elsewhere, the step size is 2. To emulate real-world scenarios with intra-transition reward sig-

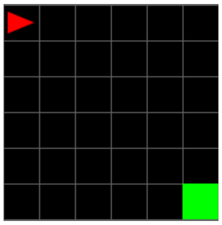

Figure 1: Minigrid 8x8 environment.

nals, we introduce a -1 reward from crossing the third row, unless we are in the rightmost column, where no negative reward exists.

5.2. MDP versus SMDP

The first experiment compares the MDP and SMDP variations in an online setting, where the agents are trained with the same exploration scheme. The learning curves (Figure 2) demonstrate that for all three algorithms, the traditional MDP variations yield a sub-optimal policy, whereas the SMDP variations all converge to the optimal policy. This is because the traditional MDP algorithms consider option-level transitions as primitive transitions. They do not account for the steps taken inside the options, thus improperly discounting the goal reward. In other words, they choose to reach the goal (mistakenly) faster at the expense of collecting the negative reward from the third row. In contrast, SMDP algorithms correctly compute the return, and as a result, choose the optimal trajectory that achieves a higher return. It is worth noting that the MDP algorithms were able to converge due to the simplicity of the example, but in general, MDP algorithms in SMDP environments may not converge at all (even to incorrect values) due to nonstationary behaviours.

5.3. Offline Learning

The next experiment compares the three SMDP variations in an offline setting. Two factors in offline RL that can pose challenges are the quality and size of the dataset. To analyze how these affect SDQN, SDDQN, and SBCQ, we create datasets consisting each of 100, 1,000, and 10,000 transitions. Trajectories in each dataset include certain percentages of optimal and sub-optimal options. We introduce two sub-optimal options that can be chosen at any step: the second-best option (25% of the time) and a random option (10%, 25%, and 50% of the time). This setup is intended to loosely resemble real-world scenarios, such as healthcare treatment management problems, where treatment decisions are not always "optimal" but are not fully random either. We train the agent using each of these datasets, and for each configuration, the network which minimizes the validation loss is used to evaluate on the test set. The test episodes are fixed across all experiments, and consists of 10 randomly selected initial starting points to test the agent's learned policy more thoroughly.

Figure 3 shows the episodic returns, averaged across the 10 test episodes for each algorithm and dataset combination. We highlight two important conclusions: first, at small dataset sizes, SBCQ greatly outperforms SDQN and SDDQN, particularly when the dataset has more optimal decisions. Second, when the buffer size increases, SDQN and SDDQN both improve performance but are at best equivalent to SBCQ. SBCQ appears to learn more robustly with a small and noisy dataset than SDQN or SDDQN. As a result, SBCQ is expected to induce a better estimate than SDQN or SDDQN for many applications of interest, such as healthcare, where datasets may be limited in size and quality.

6. Warfarin Dosing

In this section, we consider the warfarin dosing optimization setting. As described in the introduction, warfarin is an anti-coagulant commonly prescribed to patients with atrial fibrillation to reduce their risk of stroke, with dose management targeting an INR level between 2 and 3 (known as the therapeutic range).

6.1. Data

The warfarin dataset consists of data for 29,272 adult patients across four randomized control trials: ARISTOTLE (Granger et al., 2011), ENGAGE AF (Giugliano et al., 2013), ROCKET AF (Patel et al., 2011), and RELY (Connolly et al., 2009). The dataset contains records for each clinical visit, which includes the measured INR and the prescribed warfarin dose, as well as records of adverse clinical events. This is complemented by patient medical history and demographic information. We filter for patients who did not have warfarin dose or INR data, as well as for patients who had weekly doses exceeding 140mg. The final cohort, after preprocessing, consists of 28,444 patients, with trajectories that are on average 26.9 ± 13.1 steps and 604.5 ± 318.1 days.

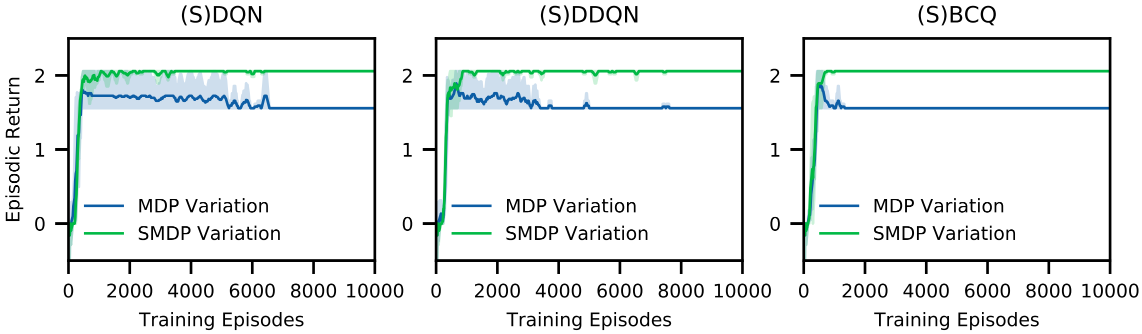

Figure 2: Episodic returns across three random seeds for (S)DQN, and (S)DDQN, and (S)BCQ.

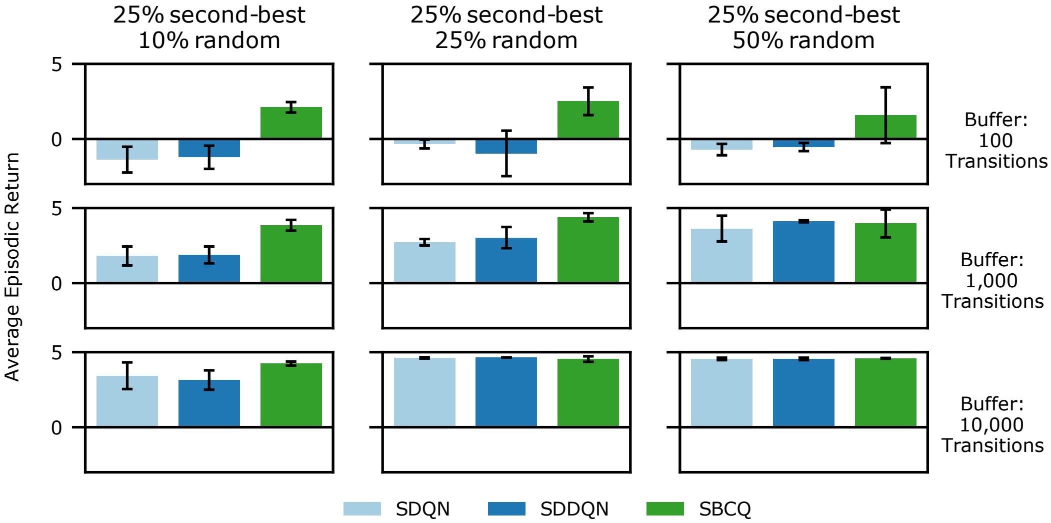

Figure 3: Episodic returns averaged across test cases for each dataset configuration and each of SDQN, SDDQN, and SBCQ. The environment here is slightly different from Figure 2 and explained in the Appendix.

The state space is comprised of the measured INR values, the prescribed warfarin dose, and the patient health indicators. The indicators are suggested by clinicians based on clinical expertise, and are sex, patient geography, medications at randomization (amiodarone, aspirin), and medical history (diabetes, myocardial infarction, hypertension, and smoking). There are seven options available to clinicians: decreasing or increasing the dose by increments of >20%, 10-20%, or 0-10%, or maintaining the current dose. As discussed in Section 4.3, we use linear interpolation to estimate the INR for each intra transition and assign a reward of +1 if the INR is within the therapeutic range of 2-3 and 0 otherwise. The ρ is the discounted return of these rewards over the intermediate time steps prior to the next observation.

To split the data, we randomly sample half of the ARISTOTLE data to create the test dataset. The validation dataset then randomly samples from the remaining ARISTOTLE and RELY data, and the rest of the data is used in training. An important metric in warfarin dosing is the time in therapeutic range (TTR). We estimate the TTR by aggregating the TTRs of the trajectories which have an agreement between the policy and clinician actions that exceed 85% of the training trajectories' agreements. The model with the highest estimated TTR on the validation dataset is chosen for evaluation on the test dataset. Technical details of data preparation and splitting are presented in the Appendix.

6.2. MDP versus SMDP

As discussed in the introduction, timing is not consistent in many practical situations, and modelling these as MDPs, with or without artificial reshaping of the data, may lead to unreliable results. The warfarin problem is a salient example. The duration between subsequent visits range from one day to three months (mean 23 ± 15 days). Importantly, the time between earlier visits may be short (1-2 days), with many transitions and option decisions during this phase. Data aggregation of weekly data points, for example, would lose the information from these entries. Hence, data aggregation must be done at a more fine-grain level, which results in unclear dynamics and numerous fake points, especially in longer visit durations (even for a single patient). Another spurious solution is to introduce time into the state space. In principle, this should make the state informative and seemingly solve the issue. However, with limited data, adding time to the state simply makes the observation space far too sparse and not learnable. We found that in practice, adding time significantly increases the training loss with almost no convergence, as expected.

Here, we examine the effect of artificial data aggregation in the warfarin setting using time steps of one day. We use linear interpolation to define the state where data is not available. This is similar to the process for computing the discounted return of an option. However, a critical distinction is that the interpolated INR is used here in both the state and reward, as opposed to the return estimation process, which only uses the results to calculate the rewards. Figure 4 illustrates how BCQ using this MDP formulation yields a policy that frequently chooses to maintain the current dose. This is due to the imbalanced option space with excessive repetition of the option "Maintain Dose" after data aggregation. The SBCQ algorithm, on the other hand, is able to learn a policy that is more nuanced and noticeably follows similar option directions as the clinician with slight preference towards increasing the dosage.

6.3. SDDQN versus SBCQ

We demonstrate that SBCQ is superior to SDDQN in the warfarin setting. In particular, we investigate the problem of over-estimation. We take the difference between the learned Q-values for both SDDQN and SBCQ against the true returns for each point along the patient's trajectory. We then normalize the difference by the true return and show the mean across all test patients for each trajectory step, illustrated in Figure 5. We limit the trajectory length to 65 steps, as more than 99.5% of the test patient trajectories satisfy this condition. Figure 5 demonstrates that A) SBCQ incurs nearly an order of magnitude lower over-estimation, and B) the over-estimation is largely consistent along steps, hence it may be less destructive in terms of recovering the optimal policy. Remark that in an ideal case, where over-estimation is a fixed-value shift, the optimal policy becomes fully recoverable.

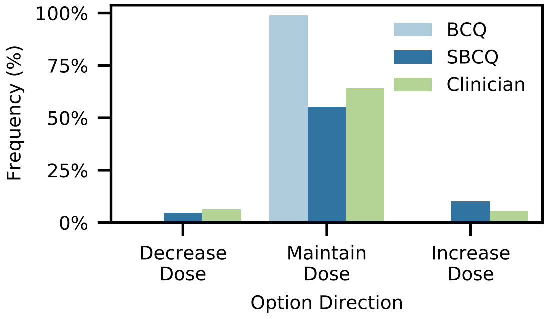

Figure 4: Options chosen by the interpolated MDP formulation of BCQ, SBCQ, and the clinicians for all test states, aggregated by option direction.

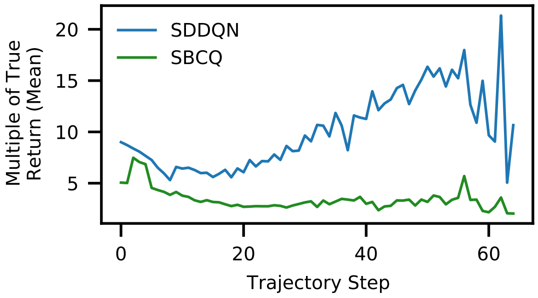

Figure 5: Deviation of SDDQN and SBCQ from the true returns (as a multiple of the true returns), averaged over test patients at each step along their trajectories.

7. Concluding Remarks

In this paper, we focused on offline RL settings with inconsistent timing, which conflate two orthogonal characteristics: being non-Markovian and being offline. We appealed to the Semi-Markov Decision Process (SMDP) theory and presented a formal methodology to modify any given value-based offline RL method to its SMDP variation. This way, we formally addressed the time-varying issue, while keeping the algorithmic machinery consistent with the contemporary offline RL literature. Consequently, with the advent of more preferable offline RL methods, we can readily translate them to their SMDP counterparts.

We further leveraged the design space that our methodology provides, and proposed three algorithms: SDQN, SDDQN, and SBCQ. To examine the resultant algorithms, we experimentally demonstrated that these algorithms converged to the optimal policy in a time-varying online environment, whereas their MDP counterparts converged to suboptimal policies. We then switched to offline settings and demonstrated that SBCQ was considerably more robust than SDQN or SDDQN, particularly when trained using smaller and/or noisier datasets, as is the case in many practical applications, such as healthcare. Finally, we presented a real-world scenario pertaining to warfarin dosing management. This problem remarkably manifests both the time-varying and offline properties. We demonstrated that the MDP framing of the problem was unable to learn a meaningful policy. Turning to the SMDP algorithms, we then showed that SBCQ was less susceptible to overestimation than SDDQN with a significant margin.

To conclude, both the formal results and our experimental investigations suggest that SMDP modelling is required in certain practical applications such as healthcare, resulting in robust and more reliable policies, especially when dealing with limited offline data.

Institutional Review Board (IRB)

Approval from the Duke University Institutional Review Board as well as institutional review boards at collaborating centers (where required) was obtained prior to receipt of any outside data. Inclusion of this element of PHI in the combined COMBINE AF is registered with PROSPERO (CRD42020178771).

Acknowledgments

We are indebted to our colleagues, who provided thoughtful insights and feedback, with special thanks to Taylor W. Killian for his advice during the project scoping process. We would also like to express our gratitude to the COMBINE AF investigators for providing the data and assisting us through any data questions. We also kindly thank the anonymous reviewers for their suggested improvements.

This research was supported in part by Microsoft Research, a CIFAR Azrieli Global Scholar Chair, a Canada Research Council Chair, and an NSERC Discovery Grant.

References

Odd O. Aalen, Ørnulf Borgan, and Håkon K. Gjessing. *Survival and Event History Analysis*. Springer New York, 2008. doi: 10.1007/978-0-387-68560-1. URL https://doi.org/10.1007/978-0-387-68560-1.

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.

Colin Bellinger, Rory Coles, Mark Crowley, and Isaac Tamblyn. Reinforcement learning in a physics-inspired semi-markov environment. April 2020.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition, 1996. ISBN 1886529108.

Steven J. Bradtke and Michael O. Duff. Reinforcement learning methods for continuous-time markov decision problems. In *Proceedings of the 7th International Conference on Neural Information Processing Systems*, NIPS'94, page 393–400, Cambridge, MA, USA, 1994. MIT Press.

Anthony P Carnicelli, Hwanhee Hong, Robert P Giugliano, Stuart J Connolly, John Eikelboom, Manesh R Patel, Lars Wallentin, David A Morrow, Daniel Wojdyla, Kaiyuan Hua, Stefan H Hohnloser, Jonas Oldgren, Christian T Ruff, Jonathan P Piccini, Renato D Lopes, John H Alexander, and Christopher B Granger. Individual patient data from the pivotal randomized controlled trials of non-vitamin k antagonist oral anticoagulants in patients with atrial fibrillation (COMBINE AF):

Design and rationale. *American Heart Journal*, 233:48–58, March 2021. doi: 10.1016/j.ahj.2020.12.002. URL https://doi.org/10.1016/j.ahj.2020.12.002.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018.

Lonnie Chrisman. Reasoning about probabilistic actions at multiple levels of granularity. 1994.

Stuart J. Connolly, Michael D. Ezekowitz, Salim Yusuf, John Eikelboom, Jonas Oldgren, Amit Parekh, Janice Pogue, Paul A. Reilly, Ellison Themeles, Jeanne Varrone, Susan Wang, Marco Alings, Denis Xavier, Jun Zhu, Rafael Diaz, Basil S. Lewis, Harald Darius, Hans-Christoph Diener, Campbell D. Joyner, and Lars Wallentin. Dabigatran versus warfarin in patients with atrial fibrillation. 361(12):1139–1151, September 2009. doi: 10.1056/nejmoa0905561. URL https://doi.org/10.1056/nejmoa0905561.

Mehdi Fatemi, Shikhar Sharma, Harm Van Seijen, and Samira Ebrahimi Kahou. Dead-ends and secure exploration in reinforcement learning. 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 1873–1881, Long Beach, California, USA, 2019. PMLR.

Mehdi Fatemi, Taylor W. Killian, Jayakumar Subramanian, and Marzyeh Ghassemi. Medical dead-ends and learning to identify high-risk states and treatments, 2021.

Vincent François-Lavet, Guillaume Rabusseau, Joelle Pineau, Damien Ernst, and Raphael Fonteneau. On overfitting and asymptotic bias in batch reinforcement learning with partial observability. *Journal of Artificial Intelligence Research*, 65:1–30, 2019.

Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms, 2019a.

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 2019b.

Joseph Futoma, Sanjay Hariharan, Katherine Heller, Mark Sendak, Nathan Brajer, Meredith Clement, Armando Bedoya, and Cara O'Brien. An improved multi-output gaussian process rnn with real-time validation for early sepsis detection. In *Machine Learning for Healthcare Conference*, pages 243–254, 2017.

Marzyeh Ghassemi, Tristan Naumann, Finale Doshi-Velez, Nicole Brimmer, Rohit Joshi, Anna Rumshisky, and Peter Szolovits. Unfolding physiological state: Mortality modelling in intensive care units. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 75–84. ACM, 2014.

Marzyeh Ghassemi, Marco AF Pimentel, Tristan Naumann, Thomas Brennan, David A Clifton, Peter Szolovits, and Mengling Feng. A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In *Proc. Twenty-Ninth AAAI Conf. on Artificial Intelligence*, 2015.

Marzyeh Ghassemi, Mike Wu, Michael Hughes, and Finale Doshi-Velez. Predicting intervention onset in the icu with switching state space models. In *Proceedings of the AMIA Summit on Clinical Research Informatics (CRI)*, volume 2017. American Medical Informatics Association, 2017.

Robert P. Giugliano, Christian T. Ruff, Eugene Braunwald, Sabina A. Murphy, Stephen D. Wiviott, Jonathan L. Halperin, Albert L. Waldo, Michael D. Ezekowitz, Jeffrey I. Weitz, Jindřich Špinar, Witold Ruzyllo, Mikhail Ruda, Yukihiro Koretsune, Joshua Betcher, Minggao Shi, Laura T. Grip, Shirali P. Patel, Indravadan Patel, James J. Hanyok, Michele Mercuri, and Elliott M. Antman. Edoxaban versus warfarin in patients with atrial fibrillation. 369(22):2093–2104, November 2013. doi: 10.1056/nejmoa1310907. URL https://doi.org/10.1056/nejmoa1310907.

Christopher B. Granger, John H. Alexander, John J.V. McMurray, Renato D. Lopes, Elaine M. Hylek, Michael Hanna, Hussein R. Al-Khalidi, Jack Ansell, Dan Atar, Alvaro Avezum, M. Cecilia

Bahit, Rafael Diaz, J. Donald Easton, Justin A. Ezekowitz, Greg Flaker, David Garcia, Margarida Geraldes, Bernard J. Gersh, Sergey Golitsyn, Shinya Goto, Antonio G. Hermosillo, Stefan H. Hohnloser, John Horowitz, Puneet Mohan, Petr Jansky, Basil S. Lewis, Jose Luis Lopez-Sendon, Prem Pais, Alexander Parkhomenko, Freek W.A. Verheugt, Jun Zhu, and Lars Wallentin. Apixaban versus warfarin in patients with atrial fibrillation. *New England Journal of Medicine*, 365(11):981–992, 2011. doi: 10.1056/NEJMoa1107039. URL https://doi.org/10.1056/NEJMoa1107039. PMID: 21870978.

Hado van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2094–2100. AAAI Press, 2016.

Katharine E Henry, David N Hager, Peter J Pronovost, and Suchi Saria. A targeted real-time early warning score (trewscore) for septic shock. *Science translational medicine*, 7(299):299ra122–299ra122, 2015.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Agata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *arXiv preprint arXiv:1907.00456*, 2019.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013.

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. *Nature medicine*, 24(11):1716–1720, 2018.

Harold Kushner and George Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer-Verlag, 2003. doi: 10.1007/b97441.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pages 45–73. Springer, 2012.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. May 2020.

Luchen Li, Matthieu Komorowski, and Aldo A Faisal. Optimizing sequential medical treatments with auto-encoding heuristic search in POMDPs. *arXiv preprint arXiv:1905.07465*, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529–533, 2015.

Robby Nieuwlaat, Lowiek M. Hubers, Alex C. Spyropoulos, John W Eikelboom, Benjamin J Connolly, Harriette G. C. Van Spall, Karleen M Schulze, Spencer M Cuddy, Alexander C. Stehouwer, Sam Schulman, and Stuart J. Connolly. Randomised comparison of a simple warfarin dosing algorithm versus a computerised anticoagulation management system for control of warfarin maintenance therapy. *Thrombosis and haemostasis*, 108 6:1228–35, 2012.

Ronald Edward Parr. *Hierarchical Control and Learning for Markov Decision Processes*. PhD thesis, University of California, Berkeley, 1998.

Manesh R. Patel, Kenneth W. Mahaffey, Jyotsna Garg, Guohua Pan, Daniel E. Singer, Werner Hacke, Günter Breithardt, Jonathan L. Halperin, Graeme J. Hankey, Jonathan P. Piccini, Richard C. Becker, Christopher C. Nessel, John F. Paolini, Scott D. Berkowitz, Keith A.A. Fox, and Robert M. Califf and. Rivaroxaban versus warfarin in nonvalvular atrial fibrillation. 365(10):883–891, September 2011. doi: 10.1056/nejmoa1009638. URL https://doi.org/10.1056/nejmoa1009638.

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. In *AMIA Annual Symposium Proceedings*, volume 2018, page 887. American Medical Informatics Association, 2018.

Munir Pirmohamed. Warfarin: The end of the end of one size fits all therapy. In *Journal of Personalized Medicine*, volume 8, 28 June 2018.

Doina Precup and Richard S. Sutton. Multi-time models for reinforcement learning. In *Proceedings of the ICML'97 Workshop on Modeling in Reinforcement Learning*, 1997.

Doina Precup, Richard S. Sutton, and Satinder Singh. Planning with closed-loop macro actions. In *In Working notes of the 1997 AAAI Fall Symposium on Model-directed Autonomous Systems*, pages 70–76. MIT Press, 1997.

Doina Precup, Richard S. Sutton, and Satinder Singh. Theoretical results on reinforcement learning with temporally abstract options. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 382–393, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

Martin Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994. ISBN 9780471619772.

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. *JMLR (Journal of Machine Learning Research): MLHC Conference Proceedings*, 2017.

Ramesh Rebba, Sankaran Mahadevan, and Shuping Huang. Validation and error estimation of computational models. *Reliability Engineering &amp; System Safety*, 91(10-11):1390–1397, 2006.

Mark Ring. Incremental development of complex behaviors through automatic construction of sensory-motor hierarchies. In Lawrence A. Birnbaum and Gregg C. Collins, editors, *Machine Learning Proceedings 1991*, pages 343–347. Morgan Kaufmann, San Francisco (CA), 1991. ISBN 978-1-55860-200-7. doi: https://doi.org/10.1016/B978-1-55860-200-7.50071-4.

Frits Richard Rosendaal, Suzanne C. Cannegieter, Felix J. M. van der Meer, and Ernest Briët. A method to determine the optimal intensity of oral anticoagulant therapy. *Thrombosis and haemostasis*, 69 3:236–9, 1993.

Suchi Saria. Individualized sepsis treatment using reinforcement learning. *Nature medicine*, 24(11): 1641–1642, 2018.

Jürgen Schmidhuber. Neural sequence chunkers. Technical report, Technische Universitat Munchen, 1991.

Sebastian Schmoll and Matthias Schubert. Semi-markov reinforcement learning for stochastic resource collection. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3349–3355. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/463. URL https://doi.org/10.24963/ijcai.2020/463. Main track.

Satinder Pal Singh. Transfer of learning by composing solutions of elemental sequential tasks. *Mach. Learn.*, 8(3):323–339, May 1992.

Samarth Sinha and Animesh Garg. S4rl: Surprisingly simple self-supervision for offline reinforcement learning. *arXiv preprint arXiv:2103.06326*, 2021.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

Harriette G.c. Van Spall, Lars Wallentin, Salim Yusuf, John W. Eikelboom, Robby Nieuwlaat, Sean Yang, Conrad Kabali, Paul A. Reilly, Michael D. Ezekowitz, Stuart J. Connolly, and et al. Variation in warfarin dose adjustment practice is responsible for differences in the quality of anticoagulation control between centers and countries. *Circulation*, 126(19):2309–2316, 2012. doi: 10.1161/circulationaha.112.101808.

Harini Suresh, Nathan Hunt, Alistair Johnson, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Clinical intervention prediction and understanding using deep networks. *JMLR (Journal of Machine Learning Research): MLHC Conference Proceedings*, 2017.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018. ISBN 9780262039246.

Richard S. Sutton. TD models: Modeling the world at a mixture of time scales. In Armand Prieditis and Stuart Russell, editors, *Machine Learning Proceedings 1995*, pages 531–539. Morgan Kaufmann, San Francisco (CA), 1995. ISBN 978-1-55860-377-6. doi: https://doi.org/10.1016/B978-1-55860-377-6.50072-4.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999. ISSN 0004-3702.

Shengpu Tang, Aditya Modi, Michael Sjoding, and Jenna Wiens. Clinician-in-the-Loop decision making: Reinforcement learning with Near-Optimal Set-Valued policies. 119:9387–9396, 2020a.

Shengpu Tang, Aditya Modi, Michael Sjoding, and Jenna Wiens. Clinician-in-the-loop decision making: Reinforcement learning with near-optimal set-valued policies. In *International Conference on Machine Learning*, pages 9387–9396. PMLR, 2020b.

C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, King's College, 1989.

Christopher J C H Watkins and Peter Dayan. Q-learning. *Mach. Learn.*, 8(3):279–292, May 1992.

World Health Organization (WHO). *World Health Statistics 2021: monitoring health for the SDGs, sustainable development goals*. Licence: CC BY-NC-SA 3.0 IGO., 2021.

Lambert E. Wixson. Scaling reinforcement learning techniques via modularity. In Lawrence A. Birnbaum and Gregg C. Collins, editors, *Machine Learning Proceedings 1991*, pages 368–372. Morgan Kaufmann, San Francisco (CA), 1991. ISBN 978-1-55860-200-7. doi: https://doi.org/10.1016/B978-1-55860-200-7.50076-3.

Mike Wu, Marzyeh Ghassemi, Mengling Feng, Leo A Celi, Peter Szolovits, and Finale Doshi-Velez. Understanding vasopressor intervention and weaning: Risk prediction in a public heterogeneous clinical time series database. *Journal of the American Medical Informatics Association*, page ocw138, 2016.

Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: a survey. *arXiv preprint arXiv:1908.08796*, 2019.

Ali Zarezade, Abir De, Utkarsh Upadhyay, Hamid R. Rabiee, and Manuel Gomez-Rodriguez. Steering social activity: A stochastic optimal control point of view. *J. Mach. Learn. Res.*, 18(1):7512–7546, jan 2017. ISSN 1532-4435.

Appendix A. Illustrative Example

This section will document additional details of the illustrative example. We use the Gym Minigrid environment (Chevalier-Boisvert et al., 2018) with an 8x8 grid, depicted in Figure 6. We use additional wrappers to modify the state space, observations, and reward structure. Most notably, we introduce options as discussed in the main body of the paper. We will release the code after the review process is concluded.

A.1. Model Architecture

We use the same fully-connected network architecture for all three algorithms: Semi Deep Q-Netowrk (SDDQN), Semi Double DQN (SDDQN), and Semi Batch-Constrained Learning (SBCQ). The state space, after converting to the flat array, has a dimension size of 108. The fully-connected network has two hidden layers, with 128 nodes and 64 nodes, followed by a linear layer which outputs the Q value of each action/option. The ReLU activation function is applied to the two hidden layers. The networks are trained using a learning rate of 0.0005 and mini-batch size of 32. For the SBCQ algorithm, we use a BCQ threshold of 0.3 for all of the experiments. Other global hyper parameters can be found in the `config.yaml` file at the root directory of the code.

A.2. Online Experiment Details

There are two reward signals in the environment: +10 when the agent reaches the goal state, and -1 when the agent crosses the third row (unless the agent is in the rightmost column). There are three possible options to take: turn left, turn right, or move forward. Each option consists of the respective primitive action, followed by a sequence of forward steps. The step size varies by location – when the agent is on the top row, the step size is 1. When the agent is in the leftmost column facing downwards, the step size is 4. Elsewhere, the step size is 2. The discount factor in the online setting is 0.9.

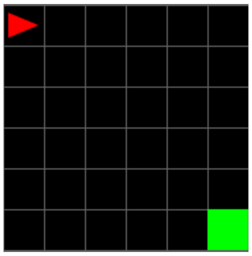

Figure 6: Minigrid 8x8 environment.

To ensure adequate exploration during training, we begin with an epsilon of 1.0 (full exploration) for 5,000 training steps, and we anneal the epsilon to 0.1 over 10,000 additional steps. After every 20 training episodes, the current policy is evaluated in the environment by running one episode with the agent starting at the default initial position (upper left corner). When the final policy is animated, we observe that the MDP-based algorithms choose the path that incurs the negative reward, whereas the SMDP-based algorithms avoid the negative reward by taking smaller steps towards the goal.

A.3. Offline Experiment Details

The results in the offline experiment are run on a slightly modified environment. The first difference is that the agent incurs a reward of -3 instead of -1 for crossing the third row. The second difference is that there are only two step sizes: a step size of 4 when the agent is in the leftmost column facing downwards, and a step size of 2 elsewhere. The third difference is that the discount factor used in the offline setting is 0.95 instead of 0.9.

To have a fully offline learning scenario, the validation dataset is offline as well. There are a few ways to select the validation data; e.g., using the same composition as the training data, or using the same validation dataset across all experiments. Regardless of the selected approach, the validation dataset is composed of 250 transitions and is used to evaluate the network under training in order to choose the best training epoch. The best network with the lowest validation loss is then selected to evaluate on the test data. In the results presented in the main text, the models were selected using the second validation approach. Figures 7 and 8 show the results using the two validation approaches. The results presented in the figures are aggregated across nine random seeds. In the final version, we will update the main text to reflect the results across more random seeds.

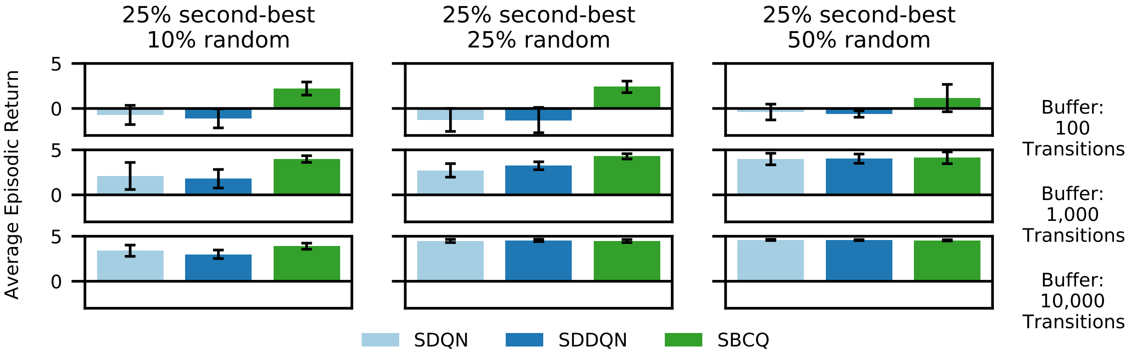

Figure 7: Episodic returns averaged across all test cases for each dataset configuration and each of SDQN, SDDQN, and SBCQ. Model selection was done using a fixed validation dataset (with 25% random decision and 25% second-best decisions) across all experiments.

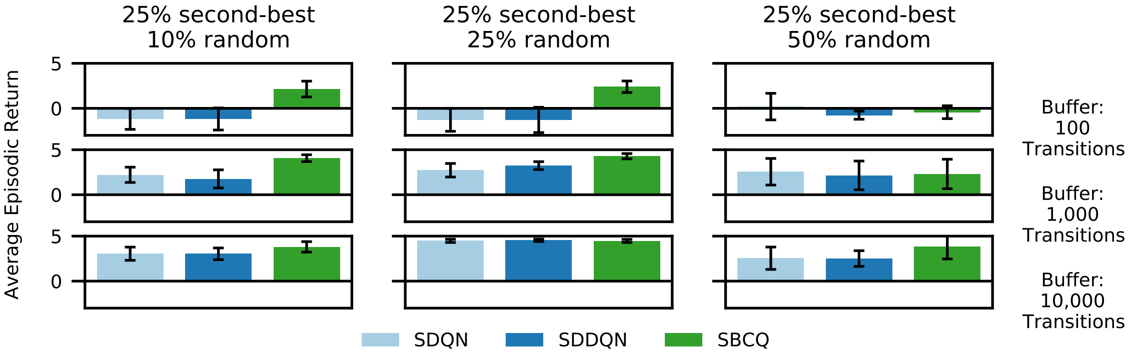

Figure 8: Episodic returns averaged across all test cases for each dataset configuration and each of SDQN, SDDQN, and SBCQ. Model selection was done using a validation dataset with the same composition as the training dataset.

Appendix B. Warfarin Dosing

Access to the warfarin data used for this study is governed by an executive committee. Details regarding the data and how to reach the committee for potential collaborations will be released after the review process is concluded.

B.1. Data Preprocessing

The warfarin dataset begins with 29,272 adult patients across the four trials. We begin by excluding patients who did not have warfarin and INR data, as well as patients who had weekly doses exceeding 140mg. This leaves us with 28,444 patients. We also remove entries that were recorded prior to the first INR measurement and after the final INR measurement.

For the state space, we extract the features deemed relevant based on clinical input. These features were age, sex, weight, region, smoking status, concurrent medications (aspirin, amiodarone, thienopyridines), and medical history (of diabetes, hypertension, myocardial infarction). The state space also consists of the current INR, the previous four INR measurements, the previous warfarin dose, and binary variables indicating whether the patient experienced ischemic stroke, hemorrhagic stroke, major bleeding, or hospitalizations during the treatment. All of the continuous features are normalized, and the categorical features are converted to one-hot encodings. To improve training, the continuous features are also discretized by one-hot encoding the values into quantiles (hence, both normalized and quantile versions are used). The fixed, demographic features and the time-varying features are summarized in Tables 1 and 2, respectively. The final state space has a dimension size of 56.

To construct the options, we use the recorded clinical decisions and discretized the dose increments based on clinical expertise and existing clinical algorithms (Spall et al., 2012; Nieuwlaat et al., 2012). Following clinical practice, the dose increments used in the underlying primitive actions are: maintain current dosage, increase in increments of <10%, 10-20%, and over 20%, or decrease in the same increments. This yields seven primitive actions. Options are then constructed as one of the seven primitive actions, followed by the primitive action of maintaining the current dosage.

There were a few cases where a patient treatment was split into multiple trajectories. The first case is when adverse medical events occur during the treatment, and the records surrounding the events may not be as accurate as other records. As such, the trajectory ends when an adverse event occurs, and the remaining treatment is treated as a new trajectory. The second case is when more than 90 days elapsed between clinical visits. Since this is a longer time elapsed than clinical recommendation (Spall et al., 2012), it is deemed that interpolation between these points (for reward estimation) is unreliable. Instead, we split this trajectory into two trajectories. After creating trajectories from the patient data, some trajectories were very short (the shortest trajectory was only one measurement). As a result, we also remove trajectories that have fewer than ten decisions.

To split the data, we split along the trials. Half of the ARISTOTLE patients were held out for testing, and the remainder of the ARISTOTLE was split between training and validation. The validation data was generated by randomly sampling from the remaining ARISTOTLE data, as well as the RE-LY data. The rest of the data was used as the training data.

B.2. Model Architectures

We use the same fully-connected network for both the DDQN and BCQ. The state space, as described in the previous section, has a dimension size of 56. The fully-connected network has two hidden layers, with 64 nodes in each, followed by a linear layer mapping to the Q value of each option. The ReLU activation function is applied to both hidden layers. The networks are trained using the learning rate of 5×10^{-5} and batch size of 64. For the SBCQ algorithm, the BCQ threshold is 0.2.

B.3. Model Validation

To select the model to use on the test data, we estimate what the time in therapeutic range (TTR) would be had the clinician followed the policy. This is estimated by evaluating the trajectories where there is good agreement between the policy and the actions chosen by the clinician in the same setting. Each of the seven options corresponds an integer as such: [0: Decrease dose by $>$ 20%, 1: Decrease dose by 10-20%, 2: Decrease dose by 10-20%, 3: Maintain current dose, 4: Increase dose by $<$ 10%, 5: Increase dose by $10 - 20\%$, 6: Increase dose by $>$ 20%]. The larger the difference between the doses, the larger the difference in the clinical outcome of the dose decision. In other words, we denote the difference between the policy and clinician decision as the absolute difference between the corresponding integers of their respective decisions. We can then sum up the differences along a trajectory, to estimate the agreement between the policy and clinician for any given trajectory. Then, we average the observed TTR for trajectories whose agreement exceeds 85% of the trajectories. This yields the estimated TTR for following the policy.

Table 1: Warfarin Demographic Features

Feature	**Data Format**
Sex	Binary Encoding
Weight	Float (Min-Max Scaling)
Weight (Discretized)	One-Hot Encoding (6 Quantile Categories)
Age	Float (Min-Max Scaling)
Age (Discretized)	One-Hot Encoding (7 Quantile Categories)
Continent	One-Hot Encoding (6 Categories)
On Aspirin	Binary Encoding
On Amiodarone	Binary Encoding
On Thienopyridine	Binary Encoding
History of Diabetes	Binary Encoding
History of Myocardial Infarction	Binary Encoding
History of Congestive Heart Failure	Binary Encoding
History of Hypertension	Binary Encoding
History of Smoking	One-Hot Encoding (3 Categories)

Table 2: Warfarin Time-Varying Features

Feature	**Data Format**
Previous Warfarin Dose	Float (Min-Max Scaling)
Previous Warfarin Dose (Discretized)	One-Hot Encoding (10 Quantile Categories)
Current INR	Float (Min-Max Scaling)
Current INR (Discretized)	One-Hot Encoding (5 Quantile Categories)
Previous Four INR Measurements	Float (Min-Max Scaling)
Minor Bleed Occurrence	Binary Encoding
Major Bleed Occurrence	Binary Encoding
Hospitalization Occurrence	Binary Encoding