ICU-Sepsis: A Benchmark MDP Built from Real Medical Data

Kartik Choudhary, Dhawal Gupta, and Philip S. Thomas
{kartikchoudh,dgupta,pthomas}@cs.umass.edu
College of Information and Computer Sciences
University of Massachusetts

Abstract

We present *ICU-Sepsis*, an environment that can be used in benchmarks for evaluating reinforcement learning (RL) algorithms. Sepsis management is a complex task that has been an important topic in applied RL research in recent years. Therefore, MDPs that model sepsis management can serve as part of a benchmark to evaluate RL algorithms on a challenging real-world problem. However, creating usable MDPs that simulate sepsis care in the ICU remains a challenge due to the complexities involved in acquiring and processing patient data. ICU-Sepsis is a lightweight environment that models personalized care of sepsis patients in the ICU. The environment is a tabular MDP that is widely compatible and is challenging even for state-of-the-art RL algorithms, making it a valuable tool for benchmarking their performance. However, we emphasize that while ICU-Sepsis provides a standardized environment for evaluating RL algorithms, it should not be used to draw conclusions that guide medical practice.

1 Introduction

In this paper, we present *ICU-Sepsis*—an easy-to-use environment that can be used in benchmarks for *reinforcement learning* (RL) algorithms. This environment is a *Markov decision process* (MDP) that models the problem of providing personalized care to sepsis patients, constructed using real-world medical records. The environment exhibits a level of complexity that challenges state-of-the-art RL algorithms, making it a suitable domain to include when benchmarking and evaluating RL algorithms. Its tabular nature makes it a lightweight and portable MDP that is compatible with many RL algorithms and which can be quickly incorporated into any benchmark suite.

Sepsis is a life-threatening condition that arises when the body's response to infection causes injury to its own tissues and organs, and requires personalized care based on a sequence of clinical decisions. This sequence of decisions results in evaluative feedback—information about whether or not the patient survived. However, this feedback does not specify what the optimal decisions would have been in retrospect, i.e., it does not provide the instructive feedback required for supervised learning (e.g., what the optimal dosages of each medicine would have been). The evaluative nature of this feedback and the potential for delays in its availability make reinforcement learning methods a natural choice for this problem.

Following the work of Komorowski et al. (2018), sepsis management has emerged as a prominent use case in applied RL research (Raghu, 2019; Yu & Huang, 2023), where historical patient data obtained from large medical record databases is used to model sepsis as an MDP. One of the most common sources of patient records is the MIMIC-III database (Johnson et al., 2023), which contains health-related data for over forty thousand ICU patients, collected between 2001 and 2012. Recognizing the widespread interest and importance of this topic, a dedicated RL environment that emulates the environments used in applied RL research for sepsis treatment in the ICU can serve as a valuable tool for evaluating the efficacy of RL algorithms for a real-world problem of interest.

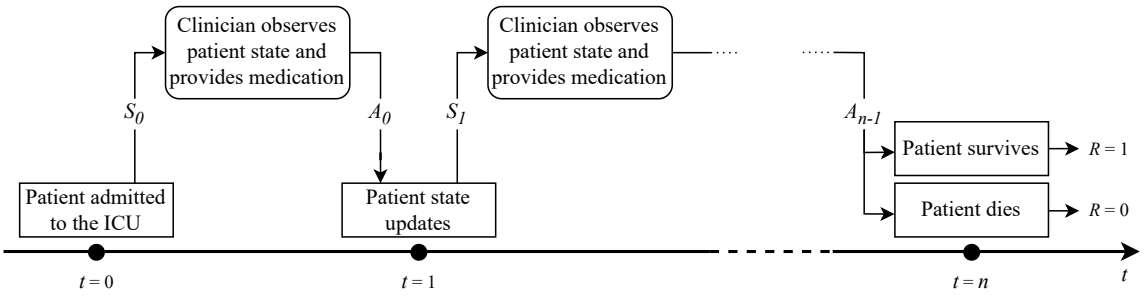


Figure 1: Illustration of one episode in the ICU-Sepsis environment. The clinician treats the patient through *actions*, which affect how their *state* evolves over time, until the patient is discharged (and a positive reward is received), or the patient dies (and no reward is received).

Various researchers have developed MDPs that simulate sepsis, as described in detail in Section 2.3. However, constructing such an MDP is a complex process of querying, cleaning, and filtering patient data from a medical database. Slight differences in the design and implementation of these procedures by different researchers have resulted in slightly different MDPs. Consequently, a standardized version of the sepsis MDP, essential for establishing a benchmark, has yet to be defined. Moreover, although the MIMIC-III database is openly available, researchers must formally request access, a process that entails completing a data protection course and signing a data use agreement. While these measures are crucial for upholding patient privacy, they, in conjunction with the complex and varying MDP creation processes, pose significant challenges for RL researchers seeking to include sepsis treatment in their benchmark suites.

ICU-Sepsis addresses these issues by presenting users with a readily deployable environment, designed for evaluating the efficacy of most RL algorithms. The MDP is a standalone environment built with the MIMIC-III database that does not require any querying, cleaning, or filtering from the user and can be used or modified without restriction (i.e., users need not complete courses or sign a data use agreement) while maintaining patient privacy (see Section 4.5 for details).

Following the precedent set by Komorowski et al. (2018), the status of a patient at any given time is discretized into a set of 716 states,[1] balancing the granularity of the state set with the amount of data available for modeling each state transition probability. Similarly, following prior work (Komorowski et al., 2018), the possible medical interventions by clinicians are discretized into 25 possible actions. The discount factor γ is set to 1 to reflect the goal of maximizing each patient's chance of survival. At the end of each episode, patient survival results in a reward of $+1$, while death corresponds to a reward of 0, with all intermediate rewards also being 0. Figure 1 shows an illustration of one episode in the ICU-Sepsis environment. An agent selecting actions uniformly randomly achieves an expected return (probability of patient survival) of 0.78, while an optimal policy computed using value iteration (Bellman, 1957) achieves an expected return of 0.88.

The ICU-Sepsis MDP is provided in a GitHub repository.[2] To allow researchers to quickly implement the environment in the software of their choice, the environment is provided as a set of CSV files containing the transition, reward, and initial state distribution matrices, as well as open-source Python implementations in OpenAI Gym (Brockman et al., 2016) and Gymnasium (Towers et al., 2023). See Section 3 for details.

[1] Komorowski et al. (2018) constructed an MDP with *roughly* 750 states. After removing some problematic states (as discussed later), and introducing additional states to model termination, the ICU-Sepsis MDP that we present contains 716 states.

[2] https://github.com/icu-sepsis/icu-sepsis

2 Background

In this section we present the notation and terminology that we use for RL, provide background regarding sepsis management, and review prior work that models sepsis treatment as an RL problem.

2.1 Technical setting

RL problems are often modeled as an agent interacting with a discrete-time Markov decision process (MDP) (Sutton & Barto, 2018; Fürnkranz et al., 2011). Formally, an MDP is a tuple of the form $(\mathcal{S}, \mathcal{A}, p, R, d_0)$, where the state set $\mathcal{S}$ contains all possible states of the environment, and the set of actions available to the agent in state $s \in \mathcal{S}$ is denoted by $\mathcal{A}(s)$. The set of all possible actions in any state is denoted by

$$\mathcal{A}^+ \doteq \bigcup_{s \in \mathcal{S}} \mathcal{A}(s).$$

In this work we consider MDPs where $\mathcal{A}^+$ and $\mathcal{S}$ are finite, unless stated otherwise. The transition function $p : \mathcal{S} \times \mathcal{A}^+ \times \mathcal{S} \to [0, 1]$ defines the probabilities of transitioning from one state to the next after taking an action: $p(s, a, s') \doteq \Pr(S_{t+1}{=}s' | S_t{=}s, A_t{=}a)$. The function $R : \mathcal{S} \times \mathcal{A}^+ \times \mathcal{S} \to [0, 1]$ gives the reward when transitioning from one state to another after taking an action. In general, this reward can be stochastic, but in our case, it is a deterministic function of S_t, A_t and S_{t+1}, written as $R_t = R(S_t, A_t, S_{t+1})$. The initial-state distribution function $d_0 : \mathcal{S} \to [0, 1]$ characterizes the distribution of the initial state: $d_0(s) \doteq \Pr(S_0 = s)$.

At any given integer time $t \geq 0$, the agent is in a state $S_t \in \mathcal{S}$, and the agent-environment interaction takes place by the agent taking action $A_t \in \mathcal{A}(S_t)$, transitioning to the next state $S_{t+1} \sim p(S_t, A_t, \cdot)$, and receiving a reward $R_t = R(S_t, A_t, S_{t+1})$. A policy $\pi : \mathcal{S} \times \mathcal{A}^+ \to [0, 1]$ defines the probability of taking each action given a state: $\pi(s, a) \doteq \Pr(A_t{=}a | S_t{=}s)$. A trajectory H of length L can be defined as a sequence of L (state, action, reward) tuples: $H \doteq (S_0, A_0, R_0, S_1, \ldots, S_{L-1}, A_{L-1}, R_{L-1})$. A dataset D is defined as a collection of such trajectories: $D \doteq \{H^{(0)}, H^{(1)}, \ldots, H^{(N-1)}\}$.

The *return* of a trajectory is the discounted sum of rewards $G(H) \doteq \sum_{t=0}^{\infty} \gamma^t R_t$, where $\gamma \in [0, 1]$ is the discount factor that determines the relative weight of future and immediate rewards. The *objective function* $J(\pi)$ is the performance measure of a policy π, defined as the expected return when the agent uses the policy π to select actions: $J(\pi) \doteq \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t\right]$. The goal of an RL agent is to find an optimal policy π^*, which is a policy that maximizes the expected return: $\pi^* \in \arg\max_\pi J(\pi)$.

2.2 Sepsis management

Sepsis is a life-threatening organ dysfunction caused by a dysregulated host response to infection (Singer et al., 2016), and is implicated in approximately 1 in every 5 deaths worldwide (Rudd et al., 2020). It is a severe multisystem disease with high mortality rates, and it is challenging to determine the correct treatment strategy for its various manifestations (Polat et al., 2017).

Sepsis management is a sequential decision-making problem, wherein clinicians make a series of medical interventions based on the state of the patients, to provide treatments that maximize the chances of patient survival. Guidelines such as those published by the Surviving Sepsis Campaign (Evans et al., 2021) provide valuable frameworks for early recognition and key interventions. However, owing to the complex nature of the condition, there are ongoing efforts to further refine guidelines and individualize treatment approaches (Kissoon, 2014; Kalil et al., 2017). In the event of a patient's death, it is generally not possible to determine the precise steps in their care that, if changed, would have resulted in their survival. Likewise, figuring out how to modify policies to enhance survival prospects for future patients remains an ongoing and critical challenge.

2.3 RL for sepsis treatment

There has been significant interest recently in the healthcare domain in using historical patient data to learn new policies for patient care, such as for diabetes (Bastani, 2014), epilepsy treatment (Pineau et al., 2009), cancer trials (Humphrey, 2017), radiation adaptation for lung cancer (Tseng et al., 2017), and many others as shown by Yu et al. (2020). In the context of sepsis management, datasets like MIMIC-III (Johnson et al., 2023) and e-ICU (Pollard et al., 2018) have been used to create tabular MDPs to find better treatment methods for sepsis (Komorowski et al., 2018; Oberst & Sontag, 2019; Tsoukalas et al., 2015; Lyu, 2020) and more specialized cases, such as pneumonia-related sepsis (Kreke, 2007), as well as optimizing the initial response to sepsis (Rosenstrom et al., 2022). Nanayakkara et al. (2022) combined distributional deep reinforcement learning (Bellemare et al., 2023) with mechanistic physiological models (Hodgkin & Huxley, 1952; Bezzo & Galvanin, 2018) to devise personalized sepsis treatment strategies. Raghu et al. (2017) studied the use of deep reinforcement learning with continuous states for optimizing sepsis treatments.

RL researchers may want to ensure that the algorithms that they develop are effective for important real-world problems like sepsis treatment. However, different (but similar) environment models are used in the applied RL research described above, and recreating these environment models can be challenging. Our work therefore seeks to provide a standardized RL environment that simulates sepsis treatment in the ICU. This environment is designed to be an easy-to-use environment within RL algorithm benchmarks, which is also representative of an important real problem. Although ICU-Sepsis is built from real data, and follows procedures from prior work intended to guide medical practice, the environment that we present is only intended for use as a standardized MDP to evaluate RL algorithms, not as a tool for studying sepsis treatment or guiding medical practice.

3 Software and Data

The dynamics of the ICU-Sepsis environment are available to download as `.csv` tables from the GitHub repository.[3] The use of `.csv` files allows for development with different libraries and programming languages. We also provide Python code compatible with the widely-used frameworks OpenAI Gym (Brockman et al., 2016) and Gymnasium (Towers et al., 2023).

3.1 The environment parameters and implementation

The states $\mathcal{S} = \{0, 1, \ldots, 715\}$ and actions $\mathcal{A}^+ = \{0, 1, \ldots, 24\}$ are both represented by integers. The transition tensor table has $|\mathcal{S}| \times |\mathcal{A}^+| = 17{,}900$ rows and $|\mathcal{S}|$ columns. The value $p(s, a, s')$ is present in the $(s')^{\text{th}}$ column of the $(s \cdot |\mathcal{A}^+| + a)^{\text{th}}$ row. The centroids of the state clusters are provided in an optional table that has $|\mathcal{S}|$ rows and 47 columns, with the s^{th} row containing the 47-dimensional centroid of state s in the normalized feature space.

The table representing the initial state distribution as a vector has 1 row and $|\mathcal{S}|$ columns. The value of $d_0(s)$ is present in the s^{th} column. The reward table also has 1 row and $|\mathcal{S}|$ columns, with the value of $R(s, a, s')$ present in the $(s')^{\text{th}}$ column. Details of reproducing these parameters from the MIMIC-III dataset are given in Appendix A.

4 The ICU-Sepsis Environment

Hospitals systematically monitor various patient statistics and vitals, documented in their *electronic health records* (EHRs) (Shabo, 2017), during the course of patient care. Clinicians prescribe appropriate medication using the collected data, adjusting dosages as the patient's condition evolves. In recent years, a growing number of hospitals have taken to recording detailed patient treatment information within their EHR systems. This rich dataset allows for the extraction of valuable insights, enabling the development of informed policies geared towards enhancing patient care.

[3] https://github.com/icu-sepsis/icu-sepsis

4.1 Formulating sepsis management as a reinforcement learning problem

Based on the statistics collected by the hospital, at any given point in time, a patient's health can be described by a vector representing different features of the patient, such as their demography, vitals, body fluid levels, etc. After discretizing time into uniform chunks, these features can be clustered into a finite set $\mathcal{S}$, thus representing the evolution of the status of a patient in the hospital as a sequence of discrete states across discrete time steps. The different types and dosages of medications administered to the patient can similarly be represented as a finite set of discrete actions $\mathcal{A}^+$. The number of different medications d_A and number of dosage levels n_A of each medication determines the size of the action set: $|\mathcal{A}^+| = (n_A)^{d_A}$.

The EHR data for $|D|$ patients can be represented as a dataset D, where each trajectory describes the hospitalization of one patient. The reward associated with each time step is $R = 0$, except for the last time step, where the reward is $R = +1$ if the patient survives. This design choice causes the expected return to correspond to the probability of a randomly selected patient surviving.

4.2 The ICU-Sepsis dataset

The dataset D is created by using real patient data describing approximately 17,000 sepsis patients from version 1.4 of the MIMIC-III dataset (Johnson et al., 2023). Following the procedure by Komorowski et al. (2018), time is discretized into 4-hour blocks, and the states are clustered using the K-means clustering algorithm (MacQueen et al., 1967) with K-means++ initialization (Arthur & Vassilvitskii, 2007). Three additional states are added to model termination—two corresponding to survival and death, based on 90-day mortality, and the third as the *terminal absorbing state* s_∞. Actions specify the dosages of intravenous fluids and vasopressors (two different interventions) with similar discretization thresholds as used by Komorowski et al. (2018).

In many states, not all actions are seen enough times to enable accurate estimation of the transition probabilities $p(s, a, \cdot)$. Therefore, for any given state-action pair (s, a), the action a is considered an admissible action for state s if and only if it occurs at least τ times in state s within the dataset, and the parameter τ is called the *transition threshold*. The set of all such admissible actions for any given state s is denoted by $\mathcal{A}(s) \subseteq \mathcal{A}^+$. Based on this definition of admissible actions, some states have no admissible actions at all, and such states are removed from the MDP.

4.3 Constructing the ICU-Sepsis MDP

Given a dataset D of trajectories, the indicator for state-action-next-state tuple (s, a, s') at time-step t in trajectory h is given by

$$I_D(h, t, s, a, s') \doteq \begin{cases} 1 & \text{if } s{=}S_t^{(h)}, a{=}A_t^{(h)}, s'{=}S_{t+1}^{(h)} \\ 0 & \text{otherwise,} \end{cases}$$

for $s, s' \in \mathcal{S}^2, a \in \mathcal{A}^+, t \in \{0, 1, \dots, \}$, and $h \in D$. This indicator is used to define the set of admissible actions $\mathcal{A}(s)$ in a given state $s \in \mathcal{S}$ as

$$\mathcal{A}(s) \doteq \left\{ a \in \mathcal{A}^+ : \sum_{h \in D, s' \in \mathcal{S}} \sum_{t=0}^{|h|-1} I_D(h, t, s, a, s') > \tau \right\}.$$

We estimate the transition probability from a state $s \in \mathcal{S}$, to another state $s' \in \mathcal{S}$, after taking an admissible action $a \in \mathcal{A}(s)$ by dividing the number of times this transition took place by the total number of times the action a was taken while in state s. Formally, the count of the number of times the transition took place is defined as $C(s, a, s') \doteq \sum_{h \in D} \sum_{t=0}^{|h|-1} I_D(h, t, s, a, s')$ and the total number of times the action was taken is defined as $C(s, a) \doteq \sum_{s' \in \mathcal{S}} C(s, a, s')$. Thus, we can define an intermediate to the transition function $\zeta : \mathcal{S} \times \mathcal{A}^+ \times S \to [0, 1]$ as $\zeta(s, a, s') = C(s, a, s')/C(s, a)$ for any admissible action $a \in \mathcal{A}(s)$, and $\zeta(s, a, s') = 0$ otherwise.

For the sake of completeness, the ICU-Sepsis environment allows every action $a \in \mathcal{A}^+$ in every state $s \in \mathcal{S}$ by defining the transition probability distribution of any inadmissible action $a \notin \mathcal{A}(s)$ to be

the average distribution for all the admissible actions in that state. The transition function for the MDP is therefore defined as

$$p(s,a,s') = \begin{cases} \zeta(s,a,s') & \text{if } a \in \mathcal{A}(s) \\ \dfrac{1}{|\mathcal{A}(s)|} \displaystyle\sum_{a' \in \mathcal{A}(s)} \zeta(s,a',s') & \text{if } a \notin \mathcal{A}(s). \end{cases}$$

This effectively means that the MDP still only allows the admissible actions to be taken, since taking an inadmissible action is equivalent to choosing one of the admissible actions at random and transitioning accordingly. Therefore, all optimal policies for the restricted-action setting remain optimal, and all policies that take inadmissible actions in some states can be mapped to equivalent policies that only use admissible actions (by spreading the probability of inadmissible actions across the admissible actions). This design decision enables the use of RL algorithm implementations that are only compatible with MDPs that allow all actions in all states, without giving them access to inadmissible actions. We discuss this decision of how inadmissible actions are handled in more detail in Appendix B.

An episode ends when the agent reaches the state corresponding to survival or death, after which it can be considered to always transition to s_∞ with probability 1 regardless of action taken. Therefore, the states corresponding to survival and death are called *terminal states*.

The policy used by the clinicians during the treatment of patients can also be estimated as $\pi_{\text{expert}}(s,a) \doteq C(s,a) / \sum_{a \in \mathcal{A}^+} C(s,a)$. The initial-state distribution d_0 is defined to be $d_0(s) \doteq \frac{1}{|D|} \sum_{h \in D} \sum_{a \in \mathcal{A}^+} \sum_{s' \in \mathcal{S}} I(h,0,s,a,s')$. The rewards are determined by the state being transitioned into, with a positive reward ($R = +1$) for transitioning into the terminal state corresponding to survival and zero reward for every other transition.

4.4 Computing the final parameters

The process of clustering the continuous state vectors into a finite set of discrete states (as mentioned in Section 4.2) introduces a source of stochasticity in the MDP parameter creation process. We investigated the effect of different seeds on the resulting MDP by creating 30 environments with different seeds (but which are otherwise identical) and analyzing their properties. We found that the different environments did not have significantly different properties, so we fixed the seed and defined the resulting MDP to be the ICU-Sepsis MDP.

The result is the transition function T represented as a tensor of shape $|\mathcal{S}| \times |\mathcal{A}^+| \times |\mathcal{S}|$, and the reward and initial-state distribution functions vectors R and d_0, respectively, both represented as vectors of length $|\mathcal{S}|$. While we have largely followed the work of Komorowski et al. (2018) in the formulation of the MDP, we have made two important changes. First, the discount factor γ has been set to 1 instead of 0.99 to prioritize patient survival over treatment speed. Secondly, the transition threshold τ has been increased from 5 to 20 to enable more accurate estimation of transition probabilities. The effects of these changes are examined in Appendix C. The values for all the parameters are shown in Table 1.

$\lvert\mathcal{S}\rvert$	d_A	n_A	$\lvert\mathcal{A}^+\rvert$	τ
716	2	5	25	20

Table 1: Parameters for creating the ICU-Sepsis MDP. The values are chosen based on work by Komorowski et al. (2018), except for τ, where the value has been increased from 5 to 20, to remove actions that are taken very rarely.

4.5 Additional environment details

The development and release of this environment has prioritized the preservation of patient privacy. The MDP parameters offer only overarching statistical summaries of patient data, which was pre-

viously de-identified during the creation of the MIMIC-III dataset. Consequently, the Institutional Review Board (IRB) review at our institution determined that the MDP and this project are exempt from IRB approval, as the research qualifies as no risk or minimal risk to subjects. Additionally, the creators of the MIMIC-III dataset affirmed the precedent of model publication derived from the dataset, provided that no straightforward method exists for reconstituting individual patient data. Therefore, the ICU-Sepsis MDP can be responsibly released, modified, and redistributed for the purposes of RL research without any substantial risk of patient harm.

	Random	Expert	Optimal	Dataset
Average return	0.78	0.78	0.88	0.77
Average episode length	9.45	9.22	10.99	13.27

Table 2: Average return and episode lengths for three baseline policies in the ICU-Sepsis MDP—a policy that takes actions uniformly randomly over all actions, the estimated expert policy, and an optimal policy computed by value iteration. The average return and episode lengths in the dataset used to create the ICU-Sepsis MDP are also shown.

Table 2 shows the baseline properties of the environment and how they compare to the MIMIC-III dataset. Since the data contains actions selected by trained physicians on real ICU patients, there are relatively few instances of poor decisions in the original dataset. This, combined with our removal of actions that were not taken at least τ times in the dataset, means that the MDP is limited to simulating reasonable treatments. If the agent selects poor or unknown treatments (actions that are inadmissible), they are mapped to a uniform distribution over the admissible (i.e., frequently selected) treatments. Hence, even an agent that selects actions uniformly randomly achieves a performance similar to that of the expert policy. However, the optimal policy computed using value iteration (Bellman, 1957) indicates that there is still room for improvement over the expert policy, which can be achieved while only taking actions that clinicians have taken in the real world.

The various design choices involved in the construction of the ICU-Sepsis environment were made with the goal of creating an easy-to-use MDP that is familiar to the RL research community. Notably, while several follow-up works have suggested improvements in the MDP creation process, like time discretization and fluid dose thresholds (see, for example, the work of Futoma et al., 2020; Tang et al., 2023), we have tried to stay generally faithful to the original design decisions made by Komorowski et al. (2018).

5 Experiments

The evaluation of RL algorithms often focuses on their ability to learn high-performing policies quickly and reliably. Hence, a good benchmark environment is one that not only resembles a real problem of interest, but one that is also challenging enough for modern algorithms that some algorithms are more effective (learn faster, converge to better policies, or learn more robustly) than others. To test the ICU-Sepsis MDP, we therefore evaluate several commonly used RL algorithms, including both value function and policy gradient methods, and analyze their learning characteristics.

Specifically, we conducted experiments to answer two research questions: **1)** How close to optimal are the policies learned using common RL algorithms? **2)** How many episodes do common RL algorithms require to find policies that perform nearly optimally?

We conduct experiments using five algorithms that represent a diverse range of approaches commonly used in RL research: Sarsa (Rummery & Niranjan, 1994), Q-Learning (Watkins & Dayan, 1992), Deep Q-Network (Mnih et al., 2013), Soft Actor-Critic (SAC) (Haarnoja et al., 2018), and Proximal Policy Optimization (PPO) (Schulman et al., 2017). We use tabular representations for the policies and value functions in all of these algorithms.

5.1 Methodology

Hyperparameter tuning is performed using a random search, where each algorithm runs for 300,000 episodes, averaged over eight random seeds for each hyperparameter setting, to maximize expected returns for the last 10% of the episodes. After approximating the best set of hyperparameters through the random search, each algorithm is run for 500,000 episodes averaged over 1,000 random seeds to ensure robustness in results. More details about the search and the final hyperparameter values are given in Appendix D. We say that an algorithm has *converged* if the average return over the last 1,000 episodes are within 0.1% of the average return over the last 10,000 episodes. Since the goal is to find policies with a high expected return in the environment, the returns are not evaluated on a separate MDP built with held-out data, as ICU-Sepsis acts as the ground truth in this case, and generalization of policies to other environments or the real world is not being tested.

5.2 Results and analysis

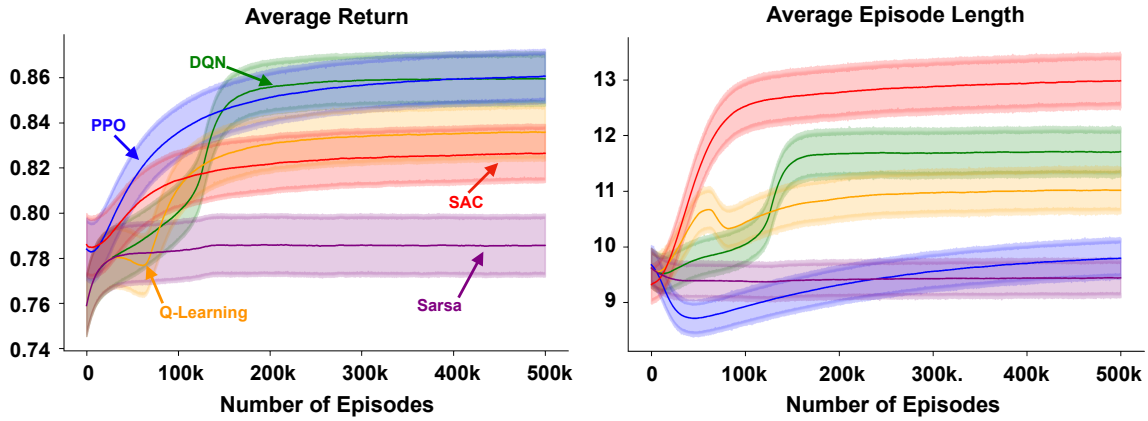


Figure 2: (**Left**) The learning curves for five algorithms on the ICU-Sepsis MDP. (**Right**) Average episode lengths during the learning process. Each curve is averaged over 1,000 random seeds, where the error bars represent one unit of standard error.

Figure 2 shows the learning curves with the average returns and average episode lengths for all five algorithms in the ICU-Sepsis environment. Table 3 shows the average number of episodes and time steps needed for each algorithm to converge.

Algorithm	Episodes (K)	Steps (M)	Average Return
Sarsa	**105.3**	**0.99**	0.79
Q-Learning	285.8	3.04	0.84
Deep Q-Network	241.5	2.60	**0.86**
SAC	324.0	4.01	0.83
PPO	386.9	3.59	**0.86**

Table 3: The number of episodes and time steps for each algorithm to converge, as well as the average return over the last 1,000 time steps. It can be observed that the algorithms require a large number of episodes to converge, and not every algorithm is able to achieve near-optimal performance.

With respect to the first research question, we observe that while some algorithms are able to achieve near-optimal performance, not all algorithms show significant improvement in performance for the learned policy, and notably, the performance of Sarsa is only marginally better than a random agent. Concerning the second research question, we observe that even after extensive parameter tuning, all of these algorithms take hundreds of thousands of episodes (i.e., millions of steps) to converge. The

average episode lengths are shown in Figure 2 (Right), which are roughly in line with the episode lengths seen in the MIMIC-III dataset, where the episodes had 13.27 steps on average.

6 Limitations

We would like to reiterate that the ICU-Sepsis MDP is designed to model a real-world problem, presenting a level of difficulty for policy search that makes it an excellent environment to evaluate RL algorithms. However, it is not intended to be a comprehensive medical simulation of sepsis and should not be used for drawing conclusions about treatments for actual patients.

Sepsis treatment requires careful consideration of numerous factors that are beyond the scope of this MDP. For example, the vasopressor dosage should change gradually, as abrupt changes can lead to hypertension or cardiac arrhythmia (Fadale et al., 2014; Allen, 2014), but basing the optimal action solely on the current state may result in policies with numerous sudden changes in vasopressor dosages, deviating from clinically accepted strategies (Jia et al., 2020). Moreover, the generalizability of the learned policies across different scenarios has not been tested, and these policies might perform suboptimally if treatment standards change over time (Gottesman et al., 2019).

7 Future Work

While ICU-Sepsis is designed to be a standardized MDP with broad compatibility with many RL algorithms, it can also serve as the base for another, more medically accurate version of the MDP that incorporates, among others things, the considerations mentioned in Section 6, making it more useful for applied RL research in the healthcare domain.

The choice of creating ICU-Sepsis as a tabular MDP is motivated by the goal of creating an MDP with broad compatibility that also reflects how RL is used in many real-world applications. As mentioned in Section 3, the normalized values of the state centroids are provided with the MDP, even though the transitions are still modeled in a tabular fashion. However, an additional continuous-state version of the MDP would further broaden the spectrum of RL algorithms that would be suitable to be evaluated on the ICU-Sepsis environment.

8 Conclusion

This work introduces the ICU-Sepsis MDP and demonstrates its potential to serve as an environment within benchmarks for RL algorithms. It is lightweight and easy to set up and use, yet the inherent complexity of the sepsis management task proves to be a significant challenge to modern RL algorithms. These qualities position the ICU-Sepsis MDP as a strong candidate for inclusion in RL benchmark suites, offering researchers an indicator of the performance of RL algorithms on an important real-world problem.

References

John M. Allen. Understanding vasoactive medications. *Journal of Infusion Nursing*, 37(2):82–86, March 2014. DOI: 10.1097/nan.0000000000000022. URL https://doi.org/10.1097/nan.0000000000000022.

David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pp. 1027–1035, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245. URL https://dl.acm.org/doi/10.5555/1283383.1283494.

Meysam Bastani. Model-free intelligent diabetes management using machine learning. 2014. URL https://era.library.ualberta.ca/items/fee1e7a7-1993-43f6-8d93-1d93855f6275.

Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2023. http://www.distributional-rl.org.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957. URL https://books.google.com/books?id=fyVtp3EMxasC.

Fabrizio Bezzo and Federico Galvanin. On the identifiability of physiological models: Optimal design of clinical tests. In *Computer Aided Chemical Engineering*, pp. 85–110. Elsevier, 2018. DOI: 10.1016/b978-0-444-63964-6.00004-0. URL https://doi.org/10.1016/b978-0-444-63964-6.00004-0.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016. URL https://arxiv.org/abs/1606.01540.

Laura Evans, Andrew Rhodes, Waleed Alhazzani, Massimo Antonelli, Craig M. Coopersmith, Craig French, Flávia R. Machado, Lauralyn Mcintyre, Marlies Ostermann, Hallie C. Prescott, Christa Schorr, Steven Simpson, W. Joost Wiersinga, Fayez Alshamsi, Derek C. Angus, Yaseen Arabi, Luciano Azevedo, Richard Beale, Gregory Beilman, Emilie Belley-Cote, Lisa Burry, Maurizio Cecconi, John Centofanti, Angel Coz Yataco, Jan De Waele, R. Phillip Dellinger, Kent Doi, Bin Du, Elisa Estenssoro, Ricard Ferrer, Charles Gomersall, Carol Hodgson, Morten Hylander Møller, Theodore Iwashyna, Shevin Jacob, Ruth Kleinpell, Michael Klompas, Younsuck Koh, Anand Kumar, Arthur Kwizera, Suzana Lobo, Henry Masur, Steven McGloughlin, Sangeeta Mehta, Yatin Mehta, Mervyn Mer, Mark Nunnally, Simon Oczkowski, Tiffany Osborn, Elizabeth Papathanassoglou, Anders Perner, Michael Puskarich, Jason Roberts, William Schweickert, Maureen Seckel, Jonathan Sevransky, Charles L. Sprung, Tobias Welte, Janice Zimmerman, and Mitchell Levy. Surviving Sepsis Campaign: International guidelines for management of sepsis and septic shock 2021. *Intensive Care Medicine*, 47(11):1181–1247, October 2021. DOI: 10.1007/s00134-021-06506-y. URL https://link.springer.com/article/10.1007/s00134-017-4683-6.

Kristin Lavigne Fadale, Denise Tucker, Jennifer Dungan, and Valerie Sabol. Improving nurses' vasopressor titration skills and self-efficacy via simulation-based learning. *Clinical Simulation in Nursing*, 10(6):e291–e299, June 2014. DOI: 10.1016/j.ecns.2014.02.002. URL https://doi.org/10.1016/j.ecns.2014.02.002.

Johannes Fürnkranz, Philip K. Chan, Susan Craw, Claude Sammut, William Uther, Adwait Ratnaparkhi, Xin Jin, Jiawei Han, Ying Yang, Katharina Morik, Marco Dorigo, Mauro Birattari, Thomas Stützle, Pavel Brazdil, Ricardo Vilalta, Christophe Giraud-Carrier, Carlos Soares, Jorma Rissanen, Rohan A. Baxter, Ivan Bruha, Rohan A. Baxter, Geoffrey I. Webb, Luís Torgo, Arindam Banerjee, Hanhuai Shan, Soumya Ray, Prasad Tadepalli, Yoav Shoham, Rob Powers, Yoav Shoham, Rob Powers, Geoffrey I. Webb, Soumya Ray, Stephen Scott, Hendrik Blockeel, and Luc De Raedt. Markov decision processes. In *Encyclopedia of Machine Learning*, pp. 642–646. Springer US, 2011. DOI: 10.1007/978-0-387-30164-8_512. URL https://doi.org/10.1007/978-0-387-30164-8_512.

Joseph Futoma, Muhammad A. Masood, and Finale Doshi-Velez. Identifying distinct, effective treatments for acute hypotension with SODA-RL: Safely optimized diverse accurate reinforcement learning, 2020. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7233066/.

Omer Gottesman, Fredrik Johansson, Matthieu Komorowski, Aldo Faisal, David Sontag, Finale Doshi-Velez, and Leo Anthony Celi. Guidelines for reinforcement learning in healthcare. *Nature Medicine*, 25(1):16–18, January 2019. DOI: 10.1038/s41591-018-0310-5. URL https://doi.org/10.1038/s41591-018-0310-5.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.

A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4):500–544, August 1952. DOI: 10.1113/jphysiol.1952.sp004764. URL https://doi.org/10.1113/jphysiol.1952.sp004764.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, Jeff Braga, Dipam Chakraborty, Kinal Mehta, and João G.M. Araújo. CleanRL: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022. URL http://jmlr.org/papers/v23/21-1342.html.

Kyle Humphrey. Using reinforcement learning to personalize dosing strategies in a simulated cancer trial with high dimensional data. 2017. URL https://repository.arizona.edu/handle/10150/625341.

Yan Jia, John Burden, Tom Lawton, and Ibrahim Habli. Safe reinforcement learning for sepsis treatment. In *2020 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE, November 2020. DOI: 10.1109/ichi48887.2020.9374367. URL https://doi.org/10.1109/ichi48887.2020.9374367.

Alistair Johnson, Tom Pollard, and Roger Mark. MIMIC-III clinical database, 2023. URL https://physionet.org/content/mimiciii/1.4/.

Andre C Kalil, David N Gilbert, Dean L Winslow, Henry Masur, and Michael Klompas. Infectious diseases society of america (IDSA) position statement: Why IDSA did not endorse the Surviving Sepsis Campaign guidelines. *Clinical Infectious Diseases*, 66(10):1631–1635, November 2017. ISSN 1537-6591. DOI: 10.1093/cid/cix997. URL http://dx.doi.org/10.1093/cid/cix997.

Niranjan Kissoon. Sepsis guideline implementation: benefits, pitfalls and possible solutions. *Critical Care*, 18(2), March 2014. ISSN 1364-8535. DOI: 10.1186/cc13774. URL http://dx.doi.org/10.1186/cc13774.

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, October 2018. DOI: 10.1038/s41591-018-0213-5. URL https://doi.org/10.1038/s41591-018-0213-5.

Jennifer E. Kreke. Modeling disease management decisions for patients with pneumonia-related sepsis. September 2007. URL http://d-scholarship.pitt.edu/8143/.

Ruishen Lyu. Improving treatment decisions for sepsis patients by reinforcement learning. March 2020. URL http://d-scholarship.pitt.edu/38498/.

James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 281–297. Oakland, CA, USA, 1967. URL https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Some-methods-for-classification-and-analysis-of-multivariate-observations/chapter/Some-methods-for-classification-and-analysis-of-multivariate-observations/bsmsp/1200512992.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning, 2013. URL https://arxiv.org/abs/1312.5602.

Thesath Nanayakkara, Gilles Clermont, Christopher James Langmead, and David Swigon. Unifying cardiovascular modelling with deep reinforcement learning for uncertainty aware control of sepsis treatment. *PLOS Digital Health*, 1(2):e0000012, February 2022. DOI: 10.1371/journal.pdig.0000012. URL https://doi.org/10.1371/journal.pdig.0000012.

Michael Oberst and David Sontag. Counterfactual off-policy evaluation with Gumbel-Max structural causal models, 2019. URL https://proceedings.mlr.press/v97/oberst19a.html.

Joelle Pineau, Arthur Guez, Robert Vincent, Gabriella Panuccio, and Massimo Avoli. Treating epilepsy via adaptive neurostimulation: a reinforcement learning approach. *International journal of neural systems*, 19(04):227–240, 2009. URL https://www.worldscientific.com/doi/abs/10.1142/S0129065709001987.

Gizem Polat, Rustem Anil Ugan, Elif Cadirci, and Zekai Halici. Sepsis and septic shock: Current treatment strategies and new approaches. *The Eurasian Journal of Medicine*, 49(1):53–58, March 2017. DOI: 10.5152/eurasianjmed.2017.17062. URL https://doi.org/10.5152/eurasianjmed.2017.17062.

Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eICU collaborative research database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1), September 2018. DOI: 10.1038/sdata.2018.178. URL https://doi.org/10.1038/sdata.2018.178.

Aniruddh Raghu. Reinforcement learning for sepsis treatment: Baselines and analysis, 2019. URL https://openreview.net/forum?id=BJekwh0ToN.

Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo A. Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment. *CoRR*, abs/1711.09602, 2017. URL http://arxiv.org/abs/1711.09602.

Erik Rosenstrom, Sareh Meshkinfam, Julie Simmons Ivy, Shadi Hassani Goodarzi, Muge Capan, Jeanne Huddleston, and Santiago Romero-Brufau. Optimizing the first response to sepsis: An electronic health record-based Markov decision process model. *Decision Analysis*, 19(4):265–296, December 2022. DOI: 10.1287/deca.2022.0455. URL https://doi.org/10.1287/deca.2022.0455.

Kristina E Rudd, Sarah Charlotte Johnson, Kareha M Agesa, Katya Anne Shackelford, Derrick Tsoi, Daniel Rhodes Kievlan, Danny V Colombara, Kevin S Ikuta, Niranjan Kissoon, Simon Finfer, Carolin Fleischmann-Struzek, Flavia R Machado, Konrad K Reinhart, Kathryn Rowan, Christopher W Seymour, R Scott Watson, T Eoin West, Fatima Marinho, Simon I Hay, Rafael Lozano, Alan D Lopez, Derek C Angus, Christopher J L Murray, and Mohsen Naghavi. Global, regional, and national sepsis incidence and mortality, 1990–2017: analysis for the global burden of disease study. *The Lancet*, 395(10219):200–211, January 2020. DOI: 10.1016/s0140-6736(19)32989-7. URL https://doi.org/10.1016/s0140-6736(19)32989-7.

G. A. Rummery and M. Niranjan. On-line Q-learning using connectionist systems. Technical Report TR 166, Cambridge University Engineering Department, Cambridge, England, 1994. URL https://www.researchgate.net/publication/2500611_On-Line_Q-Learning_Using_Connectionist_Systems.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Amnon Shabo. *Electronic Health Record*, pp. 1–6. Springer New York, New York, NY, 2017. ISBN 978-1-4899-7993-3. DOI: 10.1007/978-1-4899-7993-3_48-3. URL https://doi.org/10.1007/978-1-4899-7993-3_48-3.

Mervyn Singer, Clifford S. Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R. Bernard, Jean-Daniel Chiche, Craig M. Coopersmith, Richard S. Hotchkiss, Mitchell M. Levy, John C. Marshall, Greg S. Martin, Steven M. Opal, Gordon D. Rubenfeld, Tom van der Poll, Jean-Louis Vincent, and Derek C. Angus. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315(8):801, February 2016. DOI: 10.1001/jama.2016.0287. URL https://doi.org/10.1001/jama.2016.0287.

Jayakumar Subramanian and Taylor Killian. Sepsis cohort from MIMIC-III. https://github.com/microsoft/mimic_sepsis, 2020.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. A Bradford Book, Cambridge, MA, USA, 2018. ISBN 0262039249. URL https://ieeexplore.ieee.org/document/712192.

Shengpu Tang, Maggie Makar, Michael W. Sjoding, Finale Doshi-Velez, and Jenna Wiens. Leveraging factored action spaces for efficient offline reinforcement learning in healthcare, 2023. URL https://arxiv.org/abs/2305.01738.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL https://zenodo.org/record/8127025.

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(12):6690–6705, November 2017. DOI: 10.1002/mp.12625. URL https://doi.org/10.1002/mp.12625.

Athanasios Tsoukalas, Timothy Albertson, and Ilias Tagkopoulos. From data to optimal decision making: A data-driven, probabilistic machine learning approach to decision support for patients with sepsis. *JMIR Medical Informatics*, 3(1):e11, February 2015. DOI: 10.2196/medinform.3445. URL https://doi.org/10.2196/medinform.3445.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3–4):279–292, May 1992. ISSN 1573-0565. DOI: 10.1007/bf00992698. URL http://dx.doi.org/10.1007/BF00992698.

Chao Yu and Qikai Huang. Towards more efficient and robust evaluation of sepsis treatment with deep reinforcement learning. *BMC Medical Informatics and Decision Making*, 23(1), March 2023. DOI: 10.1186/s12911-023-02126-2. URL https://doi.org/10.1186/s12911-023-02126-2.

Chao Yu, Jiming Liu, and Shamim Nemati. Reinforcement learning in healthcare: A survey, 2020. URL https://arxiv.org/abs/1908.08796.

A Reproducing the ICU-Sepsis Parameters

The Python code for reproducing the ICU-Sepsis parameters is available in GitHub repository[4] released with this paper. Reproducing these parameters would require the researchers to download the MIMIC-III dataset from their website.[5] The initial steps for identifying patients with sepsis and extracting their features from the MIMIC-III dataset can be performed using the MATLAB code provided in the GitHub repository[6] by Komorowski et al. (2018). These steps have also been translated by Subramanian & Killian (2020) into Python scripts that produce equivalent results with minor differences. After creating the patient features, estimating the MDP parameters and creating the list of admissible actions can be done using the scripts provided in our GitHub repository. Figure 3 shows the distribution of the number of admissible actions in the states set for ICU-Sepsis.

[4] https://github.com/icu-sepsis/icu-sepsis
[5] https://physionet.org/content/mimiciii/1.4/
[6] https://github.com/matthieukomorowski/AI_Clinician

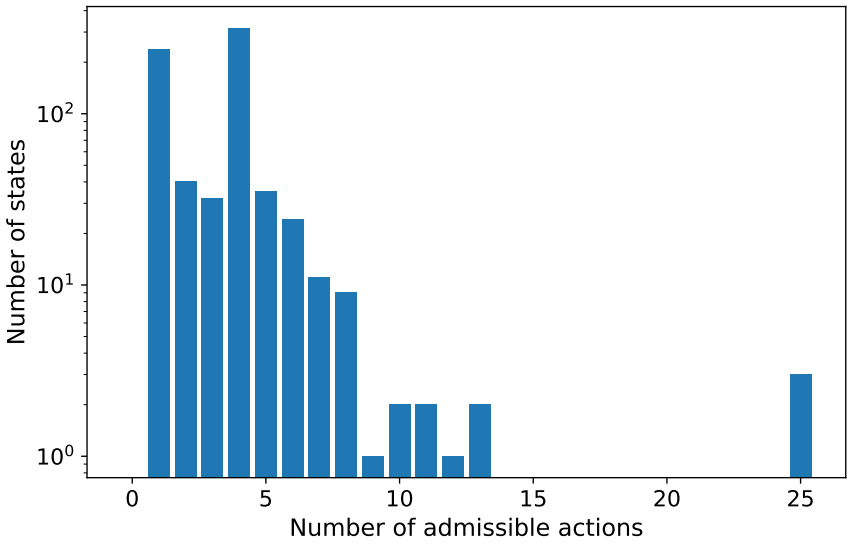


Figure 3: Distribution of the number of admissible actions for different states in the ICU-Sepsis environment.

B Handling Inadmissible Actions

Recall from Section 4.3 that state-action pairs that occur τ or fewer times in the dataset (where τ is a hyperparameter) are inadmissible (that is, they cannot be taken) since the subsequent transition distribution is unknown. This presents a problem: many implementations of RL algorithms do not allow for different action sets in each state. This may present a challenge for researchers hoping to compare to baselines that lack this functionality. We therefore opted to design ICU-Sepsis to be compatible with two different perspectives.

In the first perspective, inadmissible actions cannot be taken in the states where they are inadmissible. A list of admissible actions for each state is provided in the `extras/admissibleActions.txt` file provided with the CSV files containing the dynamics, as well as under the `admissible_actions` key in the `info` dictionary provided by the Gym/Gymnasium API. Furthermore, the entries in the transition probability table and reward function that correspond to inadmissible state-action pairs can be ignored. This perspective is ideal, simulating a setting where inadmissible actions do not exist as options for the agent to consider.

In the second perspective, we ensured that ICU-Sepsis is compatible with software that requires all actions to be admissible in all states. A key goal under this perspective was to avoid the fabrication of artificial environment behavior if inadmissible actions are chosen by the agent (e.g., defining inadmissible actions to cause a transition to a state representing death to discourage the selection of inadmissible actions). Such artificial transitions are undesirable because they can alter various performance metrics (e.g., performance improvement and learning curve plots could be dominated by the speed with which agents learn not to take inadmissible actions, which is not the important part of the ICU-Sepsis simulation). Instead we view inadmissible actions as being truly inadmissible (they cannot be taken by the agent, and hypothetical transitions that result from these actions should not be considered). To achieve this, we considered how ICU-Sepsis could be designed so that when RL software selects inadmissinble actions, these inadmissible actions are automatically modified to correspond to admissible actions, thereby ensuring that inadmissible actions are never chosen by the agent.

The key insight to enable this is the creation of a mapping from any policy that allows all actions to a corresponding policy that only selects admissible actions. Although the agent can learn and

reason using a policy that can select all actions, the interactions with the environment (including evaluations of expected return) use the corresponding policy that only selects admissible actions.

The most straightforward way to achieve these desired properties would be to define inadmissible actions to instead represent any one of the admissible actions. If there is only one inadmissible action, this essentially gives the agent two different ways to select one of the admissible actions. Critically, this does not mean that the inadmissible action is actually chosen and the simulated result is the outcome of the inadmissible action. Instead, this means that inadmissible actions can never be chosen and instead a redundant policy representation is used (a policy representation that allows for multiple ways of selecting one or more of the admissible actions).

However, this straightforward approach introduces a different issue: in standard RL implementations that require all actions to be allowed in all states, there may not be a mechanism to tell the agent that in some states two different actions actually correspond to a single action. When the agent selects one of two equivalent actions, it may not recognize that the outcome of the action provides information about both of the actions. That is, the agent will not necessarily generalize properly. This raises questions regarding the significance of the choice of *which* action inadmissible actions map to. To avoid these complexities, we opt to map inadmissible actions to a distribution over the admissible actions.

Specifically, we define inadmissible actions in a given state to be equivalent to a uniform random selection of the admissible actions in that state. This means that if an agent that requires all actions to be allowed in all states selects a inadmissible action, its policy is implicitly modified to uniformly randomly select an action from the admissible set of actions. This achieves the desired goals: it is realistic in that it completely disallows actions that it would be irresponsible to allow an RL agent to take (it does not provide hypothetical simulations of the outcomes of these uncertain and risky actions) and it avoids skewing performance metrics because the agent cannot achieve a significant initial increase in expected discounted return by simply learning to avoid inadmissible actions (a uniform random policy over all actions is now equivalent to a uniform random policy over the admissible actions). However, it is worth noting the limitation that agents selecting inadmissible actions may still fail to properly generalize, likely resulting in slower learning than agents that properly handle admissible action sets.

C Examining the Effect of the Transition Threshold

As explained in Section 4.4, the transition threshold has been increased from 5 (as set by Komorowski et al. 2018) to 20 to ensure that each admissible action is seen enough times in the dataset to provide a reasonable estimate of the transition probabilities. To examine the effects of this change on the resulting environment, we create a *Variant* environment with $\tau = 5$ that is otherwise identical to the ICU-Sepsis environment in its creation process, and ask the following research questions about the policies in this new environment: **1)** What is the highest survival rate possible in the Variant MDP? **2)** How close to the optimal performance are the policies learned by common RL algorithms? **3)** How do the average episode lengths change during the learning process for common RL algorithms?

C.1 Baseline results

Table 4 shows the baseline results for the Variant MDP and how they compare to ICU-Sepsis. We observe that an optimal policy in the Variant MDP has an expected return of 0.96, which means that 96% of sepsis patients will survive when treated using this policy, compared to the 77% survival rate seen in the MIMIC-III dataset. Thus, with respect to the first research question, the highest possible survival rate in the Variant MDP appears to be unreasonably high compared to the real data. Table 4b also shows that an episode running under this optimal policy will have an expected 24.8 steps in an episode, much higher that the 13.27 steps seen in the dataset.

Agent	ICU-Sepsis	Variant
Random	0.78	0.74
Expert	0.78	0.77
Optimal	0.88	0.96

(a) Average return

Agent	ICU-Sepsis	Variant
Random	9.45	12.6
Expert	9.22	9.8
Optimal	10.99	24.8

(b) Average episode lengths

Table 4: (**a**) Average return and (**b**) average episode lengths for ICU-Sepsis and the Variant MDP for three baseline policies: A random policy taking each action uniformly randomly in each state, the expert policy estimated from the dataset, and an optimal policy computed using value iteration. The average return and episode lengths seen in the MIMIC-III dataset were 0.77 and 13.27 respectively.

C.2 Performance of various algorithms

The number of episodes and steps required for convergence and expected returns after convergence are shown in Table 5. Figure 4 shows the learning curves and average episode lengths for the five algorithms described in Section 5 when run on the Variant MDP, using the same methodology as explained in Section 5.1 for the experiments with ICU-Sepsis.

Algorithm	Episodes (K)	Steps (M)	Average Return
Sarsa	**125.5**	**1.42**	0.79
Q-Learning	188.3	3.48	0.89
Deep Q-Network	283.3	7.82	0.91
SAC	273.3	4.20	0.87
PPO	235.5	2.35	**0.95**

Table 5: This table shows the number of episodes and time steps for each algorithm to converge, along with the average return over the last 1,000 time steps.

Therefore, with respect to the second and third research questions, we see that the expected returns and average episode lengths in the learned policies are unusually high, which do not reflect the numbers seen in the dataset. We posit that this might be happening because the agent has learned to exploit some rare actions in certain states which happened to result in good outcomes by chance. Since increasing the transition threshold removes such actions from the set of admissible actions, this behavior is not observed in the ICU-Sepsis MDP which has a higher transition threshold but is otherwise identical in the creation process to the Variant MDP. To further validate this theory, in Appendix C.3 we test the robustness of the three baseline policies: a random policy, the expert policy, and the optimal policy learned using value iteration for both ICU-Sepsis and the Variant.

C.3 Effect of perturbations on the environments

To illustrate the robustness of different policies in the ICU-Sepsis and the Variant MDP, we evaluate the performance of the baseline policies after making some perturbations in the environment dynamics. Each environment (ICU-Sepsis and the Variant) is first perturbed in the following way:

1. Among all the admissible actions, each of them is made inadmissible with some probability $\sigma \in [0, 1]$ independently of each other.
2. If all of the actions for some state are made inadmissible, one of the previously admissible actions for that state is randomly chosen and made admissible again. Thus, every state will always have at least one admissible action.
3. As explained in Section 4.3, any inadmissible action taken by the agent is equivalent to randomly choosing one of the admissible actions (according to the new list of admissible actions after the perturbation process) and taking that action.

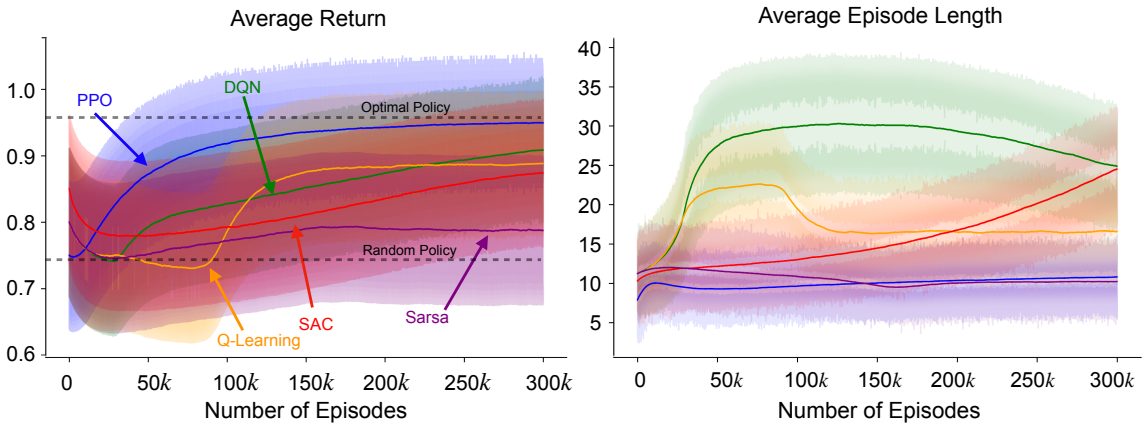


Figure 4: (**Left**) The learning curves for five algorithms on the Variant MDP. (**Right**) A plot depicting the average episode lengths during the learning process. Each curve is averaged over 20 random seeds, where the error bars represent one unit of standard error.

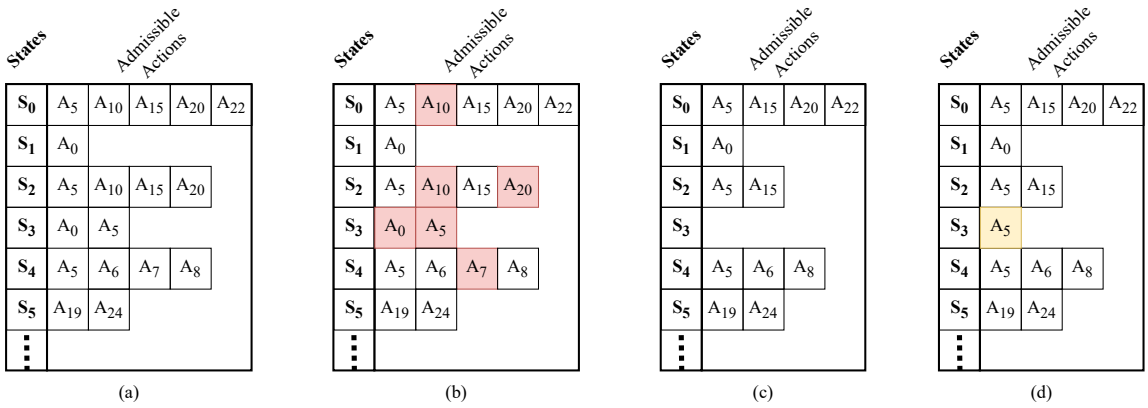


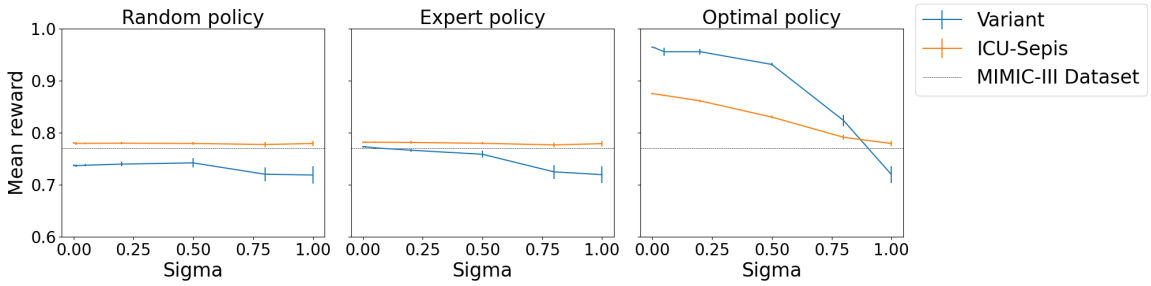
Figure 5: Illustration of the perturbation process. **(a)** Admissible actions for different states. Each row has a state (in bold) followed by the list of admissible actions in that state. **(b)** Some admissible actions are randomly chosen and made inadmissible. **(c)** Remaining admissible actions. This can cause some states (in this case S_3) to have no admissible actions left. **(d)** For states where there are no admissible actions left, a previously admissible action is chosen and reintroduced as an admissible action. Thus, every state still has at least one admissible action after the perturbation process.

Figure 5 shows an illustration of this process, which is repeated 32 times for each policy in each environment. If a policy is over-reliant on a few transitions, then their removal should result in a large performance drop. Therefore, such policies should have higher variance across runs, where some runs would not allow the actions that are being exploited to obtain unrealistically high returns. Figure 6 shows that the variance is indeed higher for the Variant compared to ICU-Sepsis, where the average return and episode lengths stay more stable as actions are progressively made inadmissible.

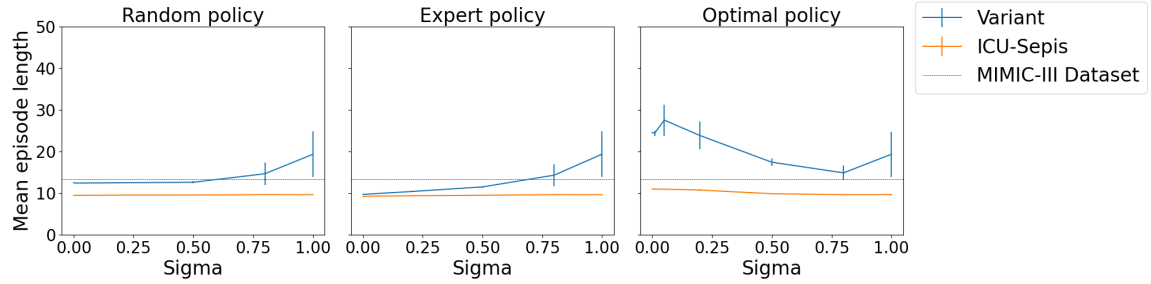
D Hyperparameters Search

The algorithms have been implemented as modifications on top of the CleanRL[7] library (Huang et al., 2022).

[7] https://github.com/vwxyzjn/cleanrl



(a) Return



(b) Number of steps per episode

Figure 6: Effects of removing some actions from the set of admissible actions on the learned policies as the probability of removing actions (σ) increases from 0 to 1. Each perturbation was done 32 times for each environment and the average and standard error of the results are shown. (**a**) The average return for different policies. (**b**) The average lengths of episodes for different policies.

D.1 Random search setting

Hyperparameter	**Value(s) / Range**	**Distribution**
Number of Seeds	8	-
Learning Rate	$[10^{-5}, 0.01]$	Log Uniform
Number of Environments	1	-
Buffer Size	$[1^3, 10^6]$	Integer Log Uniform
Discount Factor (γ)	1.0	-
Polyak Averaging Coefficient (τ)	$[0.001, 1.0]$	Log Uniform
Target Network Update Frequency	$[1, 1000]$	Integer Uniform
Batch Size	$[1, 256]$	Integer Uniform
Start Exploration Rate (ϵ_{start})	$[0.01, 1.0]$	Uniform
End Exploration Rate (ϵ_{end})	$[0.01, 0.1]$	Log Uniform
Exploration Fraction	$[0.0, 1.0]$	Uniform
Learning Starts	10,000	-
Training Frequency	10	-

Table 6: Hyperparameter settings and distribution types for the DQN hyperparameter search.

Weights & Biases (Wandb)[8] (Biewald, 2020) was utilized for performing the random search over hyperparameters. The ranges and distributions used for the searches across different algorithms are detailed in Tables 6, 7, 8, and 9. To ensure equitable compute resources across different methods, each was allocated 72 CPUs and a maximum duration of 4 days for the search, with the process concluding at that time. The number of hyperparameters explored for each method is listed in Table

[8] https://wandb.ai/

Hyperparameter	Value(s) / Range	Distribution
Number of Seeds	8	-
Learning Rate	$[10^{-5}, 0.01]$	Log Uniform
Number of Environments	1	-
Number of Steps	$[100, 500]$	Integer Uniform
Number of Mini-batches	$[1, 6]$	Integer Uniform
Discount Factor (γ)	1.0	-
GAE Lambda	$[0.0, 1.0]$	Uniform
Update Epochs	$[1, 8]$	Integer Uniform
Normalize Advantage	True	-
Clipping Coefficient	$[0.1, 0.5]$	Uniform
Clip Value Loss	True/False	-
Entropy Coefficient	$[10^{-2}, 1.0]$	Log Uniform
Value Function Coefficient	$[0.2, 1.0]$	Uniform
Maximum Gradient Norm	$[0.1, 1.0]$	Uniform
Target KL	[Null , $0.01, 0.05, 0.1$]	Uniform

Table 7: Hyperparameter settings and distribution types for the PPO hyperparameter search.

Hyperparameter	Value(s) / Range	Distribution
Number of Seeds	8	-
Learning Rate	$[10^{-5}, 0.01]$	Log Uniform
Number of Environments	1	-
Buffer Size	1	-
Discount Factor (γ)	1.0	-
Batch Size	1	-
Start Exploration Rate (ϵ_{start})	$[0.01, 1.0]$	Uniform
End Exploration Rate (ϵ_{end})	$[0.01, 0.1]$	Log Uniform
Exploration Fraction	$[0.0, 1.0]$	Uniform

Table 8: Hyperparameter settings and distribution types for the Q-learning and Sarsa hyperparameter search.

10, highlighting that slower methods were limited to fewer parameter searches. Altogether, $\geq$ 11,000 parameters were searched across all methods.

Hyperparameter	**Value(s) / Range**	**Distribution**
Number of Seeds	8	-
Buffer Size	$[10^3, 10^6]$	Integer Log Uniform
Polyak Averaging Coefficient (τ)	$[10^{-3}, 1.0]$	Log Uniform
Batch Size	$[1, 256]$	Integer Uniform
Learning Starts	$[10^4, 2 \times 10^4]$	-
Policy Learning Rate	$[10^{-5}, 0.01]$	Log Uniform
Q-function Learning Rate	$[10^{-5}, 0.01]$	Log Uniform
Update Frequency	$[1, 6]$	Integer Uniform
Target Network Update Frequency	$[100, 10^4]$	Integer Uniform
Temperature Coefficient (α)	$[0.01, 1.0]$	Uniform
Automatic Entropy Tuning	False/True	-
Target Entropy Scale	$[0.01, 1.0]$	Uniform
Number of Environments	1	-
Discount Factor (γ)	1.0	-

Table 9: Hyperparameter settings and distribution types for the SAC hyperparameter search.

Method Name	Number of Hyperparameters
Q Learning	2263
Sarsa	2501
SAC	1162
PPO	3224
DQN	2632

Table 10: Number of runs for different methods.

D.2 Best set of approximated hyperparameters

Table 11 lists the best hyperparameters for each method found during the random search. These hyperparameters were used in the experiments, with results shown in Figure 2.

Hyper-parameter	DQN	PPO	Q-Learning	SAC	Sarsa
Learning Rate	0.001	0.005	0.0025	π: 0.025, Q: 0.025	0.0025
Optimizer	Adam	Adam	Adam	Adam	Adam
Buffer Size	10,000		1	10,000	1
Batch Size	64		1	64	1
Start Exploration Rate (ϵ start)	1.0		1.0		1.0
End Exploration Rate (ϵ end)	0.001		0.001		0.001
Exploration Fraction	0.25		0.1		0.25
Learning Starts	10,000			10,000	
Training Frequency	10				
Number of Steps for Rollout		500			
Number of Minibatches		1			
GAE Lambda		0.4			
Update Epochs		6			
Normalize Advantage		Yes			
Clipping Coefficient		0.5			
Clip Value Loss		No			
Entropy Coefficient		0.005			
Value Function Coefficient		0.3			
Maximum Gradient Norm		0.4			
Target KL Divergence		0.001			
Polyak Average (τ)	0.01			0.01	
Target Network Update Frequency	512			500	
Update Frequency				1	
Alpha				0.25	
Autotune				No	
Target Entropy Scale				0.2	

Table 11: Hyper-parameters used in DQN, PPO, Q-Learning, SAC, and Sarsa to solve ICU-Sepsis.