Reinforcement Learning and Stochastic Control for Sepsis Treatment: The Promise, Obstacles and Potential Solutions

by

Thesath Nanayakkara

BSc. University of Colombo, Colombo, Sri Lanka. 2015

Submitted to the Graduate Faculty of

the Dietrich School of Arts and Sciences in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2022

UNIVERSITY OF PITTSBURGH

DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Thesath Nanayakkara

It was defended on

May 19, 2022

and approved by

Co- Committee Chair: Dr. David Swigon, Department of Mathematics, University of Pittsburgh

Co- Committee Chair: Dr. Gilles Clermont, Department of Critical Care Medicine and Department of Mathematics, University of Pittsburgh

Dr. Christopher James Langmead, Department of of Computational Biology, School of Computer Science, Carnegie Mellon University

Dr. Catalin Trenchea, Department of Mathematics, University of Pittsburgh

Reinforcement Learning and Stochastic Control for Sepsis Treatment: The Promise, Obstacles and Potential Solutions

Thesath Nanayakkara, PhD

University of Pittsburgh, 2022

We develop clinically motivated, computational methods for sepsis decision-making. Sepsis is a life-threatening syndrome, with enormous mortality, morbidity, and economic burden. However, despite decades of research spanning various academic disciplines, a thorough understanding of sepsis treatment has proved elusive.

Recent advances in data-driven machine learning and control methods have led to numerous attempts to gain insight and learn intelligent treatment strategies directly from observed data. Stochastic optimal control and Reinforcement Learning, are in particular popular as they are a natural fit to formalize clinical decision-making. However, although such methods carry significant promise, there are multiple obstacles at all levels. Thus, the goal of our work is to identify, and address these challenges and propose novel solutions. In particular, we focus on formalizing the problem in a stochastic control framework, encoding physiologic domain knowledge and improving the patient state representation, and investigating associated uncertainties.

Through a combination of control theory, deep representation learning, and the integration of mechanistic modeling we introduce several improvements and novel directions to advance the current status quo of data-driven interventions for clinical sepsis. We show how our methods can supplement clinicians, provide new directions for future computational research and *potentially* uncover valuable hints toward better treatment strategies.

Table of Contents

1.0 Introduction . . . 1

1.1 Mathematical & Machine Learning Preliminaries . . . 4

1.1.1 Reinforcement Learning, Stochastic Optimal Control, Optimization under Uncertainty . . . 4

1.1.2 Deep Learning & Representation Learning . . . 10

1.1.2.1 Deep Neural Networks: Learning & Optimization . . . 11

1.2 Sepsis And Its Patho-physiology . . . 13

2.0 Prologue to Article 0 . . . 16

3.0 Article 0: Reinforcement Learning & Stochastic Control for Sepsis Treatment: Challenges and Opportunities . . . 17

3.1 Introduction . . . 17

3.1.1 Related Work . . . 18

3.2 Background . . . 19

3.3 Reinforcement Learning & Control for Sepsis . . . 19

3.3.1 Problem Setup . . . 20

3.3.1.1 Objective & Rewards . . . 20

3.3.1.2 Partial Observability: State Representation . . . 22

3.3.1.3 Uncertainty Quantification . . . 24

3.3.2 Deep Reinforcement Learning & Algorithmic Challenges . . . 26

3.3.3 Explainability & Trustworthiness . . . 26

3.3.4 Evaluation . . . 28

3.4 Opportunities and Directions for Future Research . . . 29

4.0 Prologue to Article 1 . . . 32

5.0 Article 1: Unifying Cardiovascular Modelling with Deep Reinforcement Learning for Uncertainty Aware Control of Sepsis Treatment . . . 33

5.1 Background & Related work . . . 34

5.1.1 Reinforcement Learning . . . 34
5.1.2 Distributional & Uncertainty Aware Reinforcement Learning . . . 35
5.1.3 Reinforcement Learning in Medicine . . . 37
5.2 Results . . . 37
5.2.1 Trajectory Reconstruction Using the Physiology-driven Autoencoder 37
5.2.2 Value Distributions & Expected Values . . . 38
5.2.3 Vasopressor Treatment Strategies . . . 41
5.2.4 Uncertainty Aware Treatment . . . 43
5.2.5 Uncertainty Quantification Results . . . 45
5.2.6 A Comment on Off Policy Evaluation . . . 46
5.3 Discussion & Conclusion . . . 48
5.4 Methods . . . 50
5.4.1 Data sources & Preprocessing . . . 50
5.4.2 Models . . . 51
5.4.2.1 Physiology-driven Autoencoder . . . 51
5.4.2.2 Denoising GRU Autoencoder for Representing Lab History 54
5.4.2.3 Behavior Cloner . . . 54
5.4.3 POMDP Formulation . . . 55
5.4.3.1 Training . . . 55
5.4.4 Uncertainty . . . 56
5.4.4.1 Estimating the Uncertainty Measure . . . 57
5.4.5 Uncertainty Aware Treatment . . . 57
5.5 Supplementary Information . . . 59
5.5.1 Appendix A: Cohort Details . . . 59
5.5.2 Appendix B: Neural Network Architectures and Implementation Details 59
5.5.2.1 Physiology-driven Autoendcoder . . . 59
5.5.2.2 Denoising Lab Autoencoder . . . 60
5.5.2.3 Imitation Learning . . . 60
5.5.2.4 Bootstrapping and Deep Ensembles . . . 60
5.5.2.5 Distributional Q learning . . . 61

5.5.3 Appendix C: Additional Results 61
5.5.3.1 RL Results 61
5.5.3.2 Uncertainty Quantification Results 63
5.5.3.3 OPE Results 65
5.5.4 Appendix D: Limitations and Open Problems 66
5.5.4.1 Future Work 67
6.0 Prologue to Article 2 69
7.0 Article 2: Deep Normed Embeddings for Patient Representation . . 70
7.1 Related Work 73
7.1.1 Contrastive Learning & Representation Learning for Clinical Time Series 73
7.1.2 Reinforcement Learning for Medicine 74
7.2 Deep Normed Embeddings: Learning and Optimization 74
7.3 Results 78
7.3.1 Patient Representation on the Unit Ball 79
7.3.2 Hyper-parameter effects 82
7.3.3 Norm as a Predictor of Mortality Risk and Representation Learning for Downstream Machine Learning Tasks 85
7.3.3.1 Ablation: β and Intermediate Loss 88
7.3.4 Reinforcement Learning: Rewards and Representation 89
7.4 Discussions and Conclusions 92
7.5 Broader Impact Concerns 94
7.6 Supplementary Information 95
7.6.1 Data Sources and Preprocessing 95
7.6.2 Reinforcement Learning 96
7.6.3 Implementation Details 97
7.6.3.1 Contrastive Representation Learning 97
7.6.3.2 Baseline Representation Learning 98
7.6.3.3 Auxiliary Tasks 99
7.6.3.4 Reinforcement Learning 99

7.6.4 More Results 99
7.6.5 More RL & Control: Results and Discussions 101
8.0 Prologue to Article 3 104
9.0 Article 3: Reinforcement Learning For Survival: A Clinically Motivated Method For Critically Ill Patients 105
9.1 Related Work 106
9.2 Background 107
9.3 Reinforcement Learning for Survival 109
9.4 Experiments 112
9.4.1 Data Sources & Prepossessing 113
9.4.2 RL4S 113
9.5 Results 114
9.6 Discussions & Conclusions 117
9.7 Appendix A: Proof of Fixed Point Theorems 119
9.8 Appendix B: Stochastic Approximation Theorem 120
9.9 Appendix C: Implementation Details 121
9.10 Appendix D: RL4S: Recommended Actions 122
10.0 Conclusions 123
Appendix A. Towards a Simulated Environment Using a Deep Probabilistic Mixture of Gaussians and a Survival Model 126
Appendix B. Code Repository 128
Bibliography 129

List of Tables

1 Mean square error of reconstruction 38
2 Cohort details 59
3 RL algorithm hyper-parameters 61
4 Mean model uncertainty for survivors and non-survivors in training & validation data 65
5 Averaged relative jumps for various β 85
6 Averaged relative jumps for various intermediate loss choices, with $\beta = 0.75$ . . 85
7 AUROC for predicting if a state is t hours from death for various t 86
8 Average test AUROC for predicting if a state is t hours from death for various t 87
9 Average AUROCs for different β 88
10 Average AUROCs for intermediate loss choices- $\beta = 0.75$ in each. 88
11 Percentages of states with no treatment 92
12 Contrastive learning hyper-parameters 98
13 RL algorithm hyper-parameters 99
14 Percentage of recommended actions under different schemes and the clinician . 101
15 RL algorithm hyper-parameters 121
16 Percentages of actions (Act.) recommended by RL4S and clinicians 122

List of Figures

1 **Proposed decision support system (A)**: We use the compete patient history, which includes, vitals, scores, and labs, and previous treatment, to infer hidden states. These would all combine to make the state S_t. Our trained agent, takes this state and outputs value distributions for each treatment, its own uncertainty, and an approximate clinician's policy. We then factor in all 3 to propose uncertainty-aware treatment strategies. **The electrical analog of the cardiovascular model (B)** This provides a lumped representation of the resistive and elastic properties of the entire arterial circulation using just two elements, a resistance R and a capacitance C. This model is used to derive algebraic equations relating R, C, stroke volume (SV), filling time (T), to heart rate (F) and pressure. The Cardiac Output (CO) can be then computed as (SV)F. These equations define the decoder of the physiology-driven autoencoder. **Complete physiology-driven autoencoder network structure (C)** Patient history is sequentially encoded using three neural networks. A patient encoder computes initial cardiovascular state estimates using patient characteristics, a recurrent neural network (RNN) encodes the past history of vitals and scores, up to and including the current time point, and a transition network which takes the previous cardiovascular state, the action and the history representation to output new cardiovascular state estimates. . 36

2 Reconstruction of two validation patient trajectories using different levels of corruption using the physiology-driven autoencoder, **Left:** Heart Rate. **Right:** Systolic Blood Pressure. 38

3 Value distributions for validation patients averaged according to different times from death or discharge, **Top row**: Non Survivors. **Bottom row**: Survivors. . 39

4 Scatter plots of scaled features : **Top row**: Marker colors indicates if $\hat{V}^*(S) < 5$ (Blue) or $\hat{V}^*(S) \geq 5$ (Red) **Bottom**: Top 10 features measured by feature permutation. Here, l_k denotes the kth component of the latent lab representation. 40

5 **Top row**: Percentage of states with vasopressors recommended for the training and validation states, with time to eventual death. Here a $p\%$ voting agent, denotes an agent which only prescribes vasopressors if an only if least $p\%$ of the Bootstrapped Ensembles have agree on giving vasopressors. **Bottom row**: The percentages of states with vasopressors recommended or given with respect to cardiovascular states and SOFA score. 42

6 **Expected value evolution of the main agent for two patients: (A)** A patient who died in the ICU. **(B)** A survivor. The marker size indicates the parametric uncertainty associated with a particular action. Also shown are the standardized values of SOFA score, Systolic blood pressure, and the unidentifiable cardiovascular state (CO)R. The x-axis indicates the hours from ICU admission. **Recommended treatments under various preference parameters:** (see Eq. 28). **(C)(E)** Recommendations for the same patient as in (A). **(D)(F)** Recommendations for the same patient as in (B). **Actual clinician treatments:** **(G)** treatment for the patient in (A), **(H)** treatment for the patient in (B). . . . 44

7 **(A)** Model Uncertainty with time to death for non-survivors, **(B)** Model Uncertainty with time to discharge for survivors **(C)** Averaged entropy of value distributions for non-survivors with time to death, **(D)** Averaged entropy of value distributions for survivors with time to release. **(E)** Average Model Uncertainty for data points with density less than the p-th percentile. 47

8 Feature Importance measured by feature permutation. Here, l_k denotes the kth component of the latent lab representation 62

9 Expected Values of random validation patients, **Top**: Non-survivors, **Bottom**: Survivors. As with Fig 4, the blob size indicate the uncertainty 62

10 **L**: Heat-plots for recommended actions, under $\beta = 0.8$, $l\lambda = 0.25$ and Ensembled Distribution Expected Values. Shown are clinician's vs Agent for overall (orange), low sofa (green), medium sofa (blue), high score (purple) and non survivors last 24 hrs (red). 64

11 Box plots of validation (weighted important sampling) OPE estimates for bootstrapped ensembles 66

12 Expected Values of non-survivor, **Left**: Trained for 2 epochs, **Right** : Trained for 7 epochs 67

13 **The proposed training scheme:** We use a triplet based sampling scheme, where 3 patient states are sampled. One of them, the anchor, is always a terminal state (corresponding to death or release), and the others include a near death and a near release state. Our loss function is then defined in terms of the end result of the anchor state as shown in the figure. 75

14 **A:** Norm^2 of validation cohort non-survivors, **B:** Norm^2 of validation cohort survivors, **C:** A sample of non-survivor patient states, marked by the worst organ system 79

15 **Embedded trajectories for two non-survivors:** One patient is labeled with star markers and black/green trajectory, the second with triangle markers and orange trajectory. The marker color indicates the system with the highest organ failure score: Cardio (blue), Liver (Maroon) CNS (Purple). The first trajectory is 50 hrs long, black for the first 36 hrs, green for the last 14. The highest severity organ failure changes from cardio to CNS at 36 hrs. The embedding trajectory approaches the cluster a few hours before the organ scores indicate the change (see detail). 81

16 Embedded state distributions for various β: The labels indicate the worst organ systems as in Figure 14 82

17 Embedded state distributions without orthogonal weight initialization. Labels indicate the worst organ system 83

18 Averaged embedding norm with time to death (for non-survivors) and release (for survivors): for different β 84

19 Averaged embedding norm with time to death (for non-survivors) and release (for survivors): for various intermediate loss choices 84

20 **Results of the MLP model A:** Norm^2 of validation cohort non-survivors, **B:** Norm^2 of validation cohort survivors, **C:** A sample of non-survivor patient states, marked by the worst organ system 100

21 **Box plots of optimal values:** The results are shown for different reward schemes and representations. 102
22 **Box plots of optimal values:** The results are shown for different reward schemes 103
23 Box plots of averaged Q values: For RL4S and standard RL and stratified by patient outcome 115
24 Percentage of states with vasopressors: Recommended by RL and RL4S and administered by the clinicians 116

1.0 Introduction

Sepsis is a life-threatening inflammatory response to infection which could result in severe tissue and organ damage. Sepsis has an enormous burden in terms of mortality, morbidity and economic cost. In fact, sepsis has previously been attributed to over 6% of all hospitalizations and 35% of all in hospital deaths in the US, and an estimated economic burden of over $20 Billion per year. The outlook is not any better around the globe, with an estimated 11 million deaths per year. Treating sepsis at the ICU is very challenging due to the vast heterogeneity in septic patients at all levels: from the underlying infections, the progression of the disease, inflammatory responses, and responses to medical interventions. Moreover, despite decades of research, questions regarding vasopressor and fluid treatment have remained open. Therefore, recently there has been considerable interest in using data-driven methods to *personalize* clinical decision making and even to automate the decision making at the ICU.

Reinforcement Learning (RL) and stochastic optimal control are general frameworks for optimizing sequential decision making. RL when used alongside deep neural network based function approximators (Deep RL) has achieved superhuman performance in various domains, and at least in theory, is well suited to formalize clinical decision making at the ICU. However, leveraging modern methods to critical care medicine is far from trivial. In fact, there are significant challenges encountered at all levels.

The goal of this thesis is to identify such challenges and propose solutions from a holistic and inter-disciplinary perspective. Indeed, the efforts of optimal control theory itself have been dispersed among various mathematics, engineering, operations research, and artificial intelligence (AI) communities. The problem of treating sepsis, on the other hand, has been extensively researched by medical researchers over the past few decades with varying degrees of success. Recently, there have been various data driven methods proposed by a wide range of academic communities. Thus, we aim this work to bridge a large number of research areas, both primarily computational fields and biomedical research efforts which embrace such computational methods. The very interdisciplinary nature of the problem is a source of many obstacles and constraints the mathematical machinery that can be applied.

However, a guiding philosophy of work presented here is that the combination of traditional applied mathematics and modern machine learning methods can work in unison, and can be incredibly powerful by complementing the strengths of each other. For example, first principle mechanistic mathematical models embody decades of medical and physiological knowledge, and using such models can amongst other things improve explainability, trustworthiness and provide some causal inference in AI systems. Uncertainty quantification can tell us when and when not to be confident on the results of such a system.

The main content of the thesis is presented via four articles. Before each article, we have included a prologue chapter that discusses how the article fits in to the greater scope of the text. However, the articles themselves are presented almost exactly [1] as they were published or submitted for publication, thus do contain some overlap.

The first article (Article 0) is an abridged version of a review article, which discusses the stochastic control problem and the various efforts which have been made to make it more amenable to formalize intensive care decision making. This article also presents a detailed discussion of challenges and the potential of RL, which motivated the work that follow. However, we omitted some sections of this article as they are mentioned elsewhere in the thesis.

The next article presents our efforts to improve the patient representation using a relevant physiological cardiovascular model. In particular, we show how first principle based mechanistic models and modern deep learning methods can work together to provide physiologically meaningful representations from EHR data; and how such a representation can be used to combat the problem of partial observability of the state. This article also presents our uncertainty quantification efforts, focused on both aleatoric and epistemic uncertainties. Finally, we present a simple framework for uncertainty aware decision making with human clinicians in the loop and compare the recommendations from the RL agent with observed clinicians' actions.

The third article is based on a novel contrastive optimization method that has multiple benefits even beyond RL. Here, we encode high dimensional patient states to a lower dimensional unit ball such that patients with the same mortality risk are mapped to the same

[1] With the exception of Article 0 which is intended as a review article.

level set (with respect to the embedded norm). Further, we show how the learned method can map different physiological causes (organ failures) to different parts of the sphere, thus creating an embedded space that is aware of both mortality risk and organ failure. The norm of the learned embedded space can be taken as a mortality risk score. Therefore, that work also presents a systematic method of defining rewards for the RL problem, which is one of the most challenging issues faced when using RL for critical care applications.

In the fourth article, we present a novel clinically motivated control objective for critically ill patients. We then refine this objective to a practical Deep Reinforcement Learning algorithm, which works with any value based Deep RL algorithm with one line modification. We show our method has the same theoretical guarantees as Q learning and then empirically show how this method results in clinically intuitive results.

In summary, major contributions of this work include:

- Discussing the inherent challenges of applying *any* mathematical or computational method to control sepsis treatment, and potential solutions from an unified perspective.
- Introducing a way to encode physiological knowledge by a novel unsupervised learning method we call *physiology driven autoencoder*.
- Formalizing parametric uncertainty in offline RL and proposing a simple computational method to estimate this quantity for Deep Reinforcement Learning methods. We then propose a framework for uncertainty aware decision making with humans in the loop.
- Introducing a novel semi-supervised, contrastive method to embed high dimensional EHR data in a lower dimensional unit closed ball. By using simple geometric priors this embedding is aware of both mortality risk and underlying physiological causes. We also propose a method to define rewards for RL systematically and empirically show how the resulting policies could depend on the reward choice.
- Introducing a novel control objective and Deep RL method called *RL4S: Reinforcement Learning for Survival*. This objective is naturally suited for critically ill patients. We then discuss several alternate interpretations of this method, resulting in a simple RL algorithm. Further, we present theoretical results and conduct various experiments.

1.1 Mathematical & Machine Learning Preliminaries

1.1.1 Reinforcement Learning, Stochastic Optimal Control, Optimization under Uncertainty

Sequential decision making under uncertainty is ubiquitous in all forms of human activity. Naturally, this problem has been examined in detail by numerous research communities. Indeed, the terms Reinforcement Learning (RL), Stochastic Control, Sequential Optimization all describe classes of methods for decision making such that some objective will be optimized over some horizon [116, 15, 95, 21, 85, 75].

We will start by defining the stochastic control problem in the most abstract form, as defined in [15], and later discuss the more specific case which is more common in recent RL literature. However, we will focus exclusively on the discrete time case.

Our presentation here strongly follows [15], with minor changes in notation and definitions to be consistent with modern RL literature.

We will use the following notations and definitions throughout this presentation.

- $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces.
- For every $s \in \mathcal{S}$ there exists, a set (possibly $\mathcal{A}$ itself) $\mathcal{A}(s) \subseteq \mathcal{A}$, which is the control constraint.
- A policy π is a (possibly stochastic) mapping from $\mathbb{N} \times \mathcal{S}$ to $\mathcal{A}$, such that $\pi(t, s) \in \mathcal{A}(s)$, for all $s \in S.$ and $t \in \mathbb{N}$. When the policy does not depend of time, we call the policy stationary. Such a policy can be considered as function from $\mathcal{S}$ to $\mathcal{A}$. Our focus will be on stationary policies, from this point.[2] We will use Π to denote the class all stationary policies.

We will further borrow the following notations from [15], to define the abstract problem.

- Let F be the set of all extended real valued functions $J : \mathcal{S} \to \mathbb{R}^*$ or $J : \mathcal{S} \times \mathcal{A} \ \to \mathbb{R}^*$
- Let B be the Banach space of all bounded real valued functions $J : \mathcal{S} \to \mathbb{R}$, and this case $||J||$ is the standard sup-norm. i.e. $||J||= \sup_{s\in\mathcal{S}} J(s)$

[2]This does not sacrifice any generality as the state can be defined to include time, so any policy is a stationary policy with respect to the new state space.

Next, we have a given mapping

$$H : \mathcal{S} \times \mathcal{A} \times F \to \mathbb{R}^* \tag{1}$$

Informally, H can be interpreted as a relationship between the total cost/reward at the N^{th} stage, and the $(N+1)^{\mathrm{st}}$ state. Now, for a fixed policy $\pi \in \Pi$, we define the mappings $T_\pi, T : F \to F$ by:

$$T_\pi(J)(s) = H(s, \pi(s), J) \tag{2}$$

and

$$T(J)(s) = \sup_{\pi \in \Pi} H(s, \pi(s), J) \tag{3}$$

The following monotonicity property for H is assumed. $\forall \pi, J, J' \in F$:

$$J(s) \leq J'(s) \quad , \forall s \quad \implies H(s, \pi(s), J) \leq H(s, \pi(s), J') \tag{4}$$

Now we are ready to define the formal control objective. Suppose we have a given real valued function J_0, defined on $\mathcal{S}$ (or on $\mathcal{S} \times \mathcal{A}$). For some $N \in \mathbb{N}$, let's define the N stage (episodic) *reward* function [3]. associated with the policy π as

$$J_{N,\pi}(s) = T_\pi^N(J_0)(s) \tag{5}$$

Here, T_π^N is T_π composed with itself N times.

Similarly, we can define the infinite horizon reward function as:

$$J_\pi(s) = \lim_{k \to \infty} T_\pi^k(J_0)(s) \tag{6}$$

Now we define our N stage problem as (for any $s \in \mathcal{S}$):

$$\text{maximize} \quad J_{N,\pi}(s) \quad \text{such that} \quad \pi \in \Pi, \tag{7}$$

And its analogous infinite horizon problem as (again for any $s \in \mathcal{S}$):

$$\text{maximize} \quad J_\pi(x) \quad \text{such that} \quad \pi \in \Pi \tag{8}$$

(Provided the limit exists)

[3]Note that we have not used any definition of rewards as yet, we have changed the terminology from cost to reward to be consistent with RL terminology

For a fixed $s \in \mathcal{S}$, we denote the corresponding the optimal reward functions by, J_N^* and J^*. A policy $\pi^* \in \Pi$, is said to be optimal for the two problems if $J_{N,\pi^*} = J_N^*$ and $J_{\pi^*} = J^*$ respectively. A policy is said to be uniformly N optimal if the policy is $N - i$ optimal for all $i = 0, 1, ...N - 1$.

Under this setting, it can be shown that under some mild assumptions, optimal reward functions and optimal policies can be computed from a Dynamic Programming (DP) like method. That is for the N stage problem as $J_N^* = T^N(J_0)$ for an appropriate J_0, and for the infinite state problem as $J^* = T^*(J^*)$. We refer the reader to [15] for more details and measurability concerns for uncountable probability spaces. This general setting subsumes the deterministic optimal control problem, stochastic optimal problem, among many others. However note that this formulation typically requires the knowledge of the environment, but there exist some iterative algorithms such as Q Learning, which we will describe later in the context of RL.

We will now return to the more familiar setting of RL. There, J will represent a *value function* or a *value distribution*, defined in terms of rewards and T, T_π will be appropriate Bellman operators. We will return to the abstract setting when we discuss alternate objectives for the RL problem for sepsis. In addition, we believe the abstract stochastic control problem can provide valuable insight for future algorithmic design. We will also briefly compare the RL problem with the abstract setting at the end of this section.

RL can be formalized by a Markov Decision Process (MDP) framework. The state and action spaces $\mathcal{S}$ and $\mathcal{A}$ are the same as above. In addition, there exists a (typically unknown) Markov probability kernel $p(|s, a)$, which gives the dynamics of the next state, given the current state and the action and a reward process with a kernel $r(|s, a)$.

Given a discount factor $\gamma \in (0, 1]$, the return is defined as the cumulative discounted rewards : $\sum_{t=1}^{\infty} \gamma^t r_t$, which is a random variable. In RL, the agent's performance is measured in terms of the return, and typically most of the attention has been focused on the *expected* return.

Therefore, the *value* of a policy π at state s ($V^\pi(s)$), is defined as the expected future rewards starting from state s, and following the policy π. That is :

$$V^\pi(s) = \mathbb{E}_{p,\pi}[\Sigma_t \gamma^t r_t | s_0 = s, \pi], \quad \forall s \in \mathcal{S} \tag{9}$$

The Bellman equation for the value function can be written as:

$$V^{\pi}(s) = \mathbb{E}_{p,\pi}[r + \gamma V^{\pi}(s')], \tag{10}$$

If V^* is the optimal value function, V^* satisfies the following Bellman optimality equation:

$$V^*(s) = \sup_{\pi \in \Pi} \{\mathbb{E}_{p,\pi}[r + \gamma V^*(s')\} \tag{11}$$

Similarly the *state action value function* or Q function can be defined as:

$$Q^{\pi}(s,a) = \mathbb{E}_{p,\pi}[\Sigma_t \gamma^t r_t | s_0 = s, \pi, a_0 = a], \quad \forall s \in \mathcal{S}, a \in \mathcal{A} \tag{12}$$

The Q function can be interpreted as the expected return, when starting at state s, taking the action a, and then following the policy π. Then, the following can then be verified.

The Bellman equation for the Q function:

$$Q^{\pi}(s,a) = \mathbb{E}_p[r] + \gamma \mathbb{E}_{p,\pi}[Q^{\pi}(s',a')], \tag{13}$$

[4]

and the Bellman optimality equation for the Q function:

$$Q^*(s,a) = \mathbb{E}_p[r] + \gamma \mathbb{E}_p[\sup_{a' \in A} Q^*(s',a')] \tag{14}$$

(where $Q^*(s,a)$ is the optimal Q function, and s' denotes the random next state).

We can notice that equations 10 and 13, are analogous to 2 in our abstract control problem with $H : \mathcal{S} \times \mathcal{A} \times F \to \mathbb{R}^*$ as $\mathbb{E}_p[r(s,\pi(s)] + \gamma \mathbb{E}_{p,\pi}[J(s')]$ and $\mathbb{E}_p[r(s,a)] + \gamma \mathbb{E}_{p,\pi}[J(s',\pi(s')]$ respectively for the two cases. Similarly, for finite action spaces, equations 11 and 14 are analogous to 3, with the same function H.

Indeed, it can be shown that under some regularity conditions all four Bellman operators are contractions in L^{∞}. So an iterative algorithm would converge to either optimal value functions or the policy induced value function. Value iteration and policy iteration are two such dynamic programming algorithms.

[4]Note that here, r is a random variable distributed according to an appropriate conditional distribution, conditioned on both s and a. Whilst r in 10 is only conditioned on the state and the policy (If the policy is deterministic the interpretation becomes similar). However, we don't make the conditioning explicit in our notation, for simplicity.

However, note that until now we have assumed the knowledge of the environment. As we mentioned earlier in RL, we usually do not have access to the underlying probability kernel nor the reward process. Therefore the goal of RL is to learn from *experience* which typically consists of observed transitions of the form (s_t, a_t, s_{t+1}, r_t). Therefore, there are numerous algorithms that do not assume environment dynamics [116]. These include Monte Carlo methods and temporal difference methods. Monte Carlo methods are the simplest, and these methods estimate a policy induced value function, by averaging the objective values generated from multiple trajectories starting at a state and following a policy. However these methods are typically *on-policy*, which means in order to estimate a value of a given policy, the data must be collected by following the same policy.

Temporal difference (TD) methods leverage Equations 10, 13 and 14 to derive incremental, stochastic approximation based algorithms. For example, TD (0) uses Equation 10 to estimate V^π, for a fixed policy π. More specifically, given an experience tuple (r_t, s_t, s_{t+1}, a_t) where a_t is sampled with respect to π. We can then define the temporal difference (TD) error δ_t as $\delta_t := r_t + \gamma V^\pi(s_{t+1}) - V^\pi(s_t)$. Notice, that δ_t is a sample based estimator of the difference between left and right hand sides of Equation 10. Then using an initial estimate of $V^\pi(s)$ for all states s, the following incremental algorithm can be derived.

$$V^\pi(s) \leftarrow V^\pi(s) + \alpha_t \mathbb{I}_{\{s=s_t\}}(\delta_t) \tag{15}$$

Where $\alpha_t \in (0, 1]$ is the step size. Notice that at each iteration the value function estimate only changes at $s = s_t$. For finite state MDPs, we can use a look up table to represent the approximate values for all states and assuming all states are visited infinitely often, this algorithm converges if α_t satisfy the Robbins-Monro conditions [106] : $\Sigma_{t=0}^{\infty}\alpha_t = \infty$ and $\Sigma_{t=0}^{\infty}\alpha_t^2 < \infty$

Analogously, *Q Learning* [127] uses an incremental algorithm motivated by Equation 14, which learns the optimal state, action value function directly without learning any policy induced value. Similarly to above if we observe an experience tuple of the form (r_t, s_t, s_{t+1}, a_t) (where now a_t can be any action, and thus Q learning is off-policy), we can define δ_t as :

$$\delta_t = [r_t + \gamma \text{max}_{a' \in \mathcal{A}} Q^*(s_{t+1}, a')] - Q^*(s, a)$$

Then the incremental algorithm uses the following update :

$$Q^*(s,a) \leftarrow Q^*(s,a) + \alpha_t \mathbb{I}_{\{s=s_t, a=a_t\}}(\delta_t) \tag{16}$$

Again, for the discrete state space, if all states are visited infinitely often the same convergence properties as TD(0) hold.

However, in practice state spaces are high dimensional and continuous. Therefore it is common to parameterize the Q function or the V function, by a function approximator [116]. The parameters are then learned with the intention of minimizing the td-errors (δ_t). Examples include *Fitted Q iteration*, which could be used with parametric or non-parametric function approximators [21]. Depending on the class of function approximators contraction based arguments can be used to deduce convergence.

Deep Reinforcement Learning (Deep RL) can be defined as the set of methods that use a (deep) neural network to parametrize the value function [5]. In Deep RL typically, several heuristic tricks [87] are used to help performance and the theoretical foundations of convergence are less understood.

Distributional Reinforcement Learning [11] considers the whole distribution of the return rather than focusing on just the expected value. Using the notation used in [11], we will denote the random return following a policy π at a state s by G^π.

That is :

$$G^\pi(s) = \Sigma_t \gamma^t r_t(s_t, a_t), \text{ where } s_0 = s$$

Then they show that the analogous distributional Bellman equation:

$$G^\pi(s) =_d r + \gamma G^\pi(S') \tag{17}$$

holds. Where S' is a random variable distributed according to transition dynamics $p(|s,a)$ and $_d$ denotes distributional equivalence. (i.e. both random variables on the left hand side and the right hand side have the same distribution function).

[5] There are alternate parameterizations including policy optimization: where a parametric function can represent a policy and then directly optimize the policy using sampled returns and actor critic methods : which is a combination between value based methods and policy based methods.

Policies and Risk Sensitive Reinforcement Learning: In more traditional RL, given the value function or the Q function the optimal actions were defined as the actions which maximize Q or V.

For example, using the Q function the optimal action a^* at state s can be computed as $a^* = \text{argmax}_{a \in \mathcal{A}} Q(s, a)$, or using the value V the optimal policy π^* can be computed as :

$$\pi^* = \arg\max_{\pi \in \Pi} V^\pi = \arg\max_{\pi \in \Pi} \mathbb{E}^\pi[G^\pi(s_0)] \tag{18}$$

where s_0 is the initial state. (Here we implicitly assume the maximum exists in the class of policies or actions).

Using distributional methods, we can now replace the expectation operator in Equation 18 with a general risk measure ρ acting on the random return random variable G^π. Then the policy can instead be defined as.

$$\pi^* = \arg\max_{\pi \in \Pi} \rho(G^\pi(s_0)) \tag{19}$$

Some possible risk measures include a) value at risk $\rho^\tau_{VAR}(Z) = F_Z^{-1}(\tau)$. (Where F_z is the cumulative distribution function of a random variable Z), and b) conditional value at risk : $\rho^\tau_{CVAR}(Z) = \mathbb{E}[Z|Z < [F_Z^{-1}(\tau)]$.

1.1.2 Deep Learning & Representation Learning

Deep Learning [44] is a set of computational tools, which were originally inspired by the neural structure of human brains (Thus, these methods were historically called artificial neural networks). Although the analogy between human neural circuits has weakened, deep neural networks have achieved incredible, sometimes super-human success in various tasks [114, 35, 20, 57, 23, 100], and are becoming an indispensable tool across all computational science fields.

Mathematically, a deep neural network is just a composition of differentiable (or almost everywhere differentiable) functions parameterized by a set of parameters. Modern neural networks can have billions of different parameters and these parameters are tuned by stochastic

gradient based optimization methods (such as [62]). Informally, deep learning methods use a hierarchy of ordered abstract representations, with each representation building on previous representations. The success of deep learning can be attributed to learning these intelligent representations with minimal or no human intervention: in contrast to more traditional machine learning methods that typically require a carefully prepared feature representation. Representation Learning [13, 44] is the use of machine learning to learn an informative representation, usually with the intention of using the learned representation for a downstream task.

As a subset of Machine Learning, Deep Learning methods are typically categorized into supervised and unsupervised learning, where supervised learning refers to problems where the learning task can be framed as learning a mapping between input data and labels (targets). However, *Self-Supervised Learning* [73] has received considerable attention in recent times. Self-supervised methods do not depend on labeled data [6], but attempt to obtain the supervision by exploiting some underlying structure of the data automatically. An example of a common self-supervised learning method in Natural Language Processing (NLP) is to train a network to predict a hidden part of a sentence using the remaining part. The idea is that by performing the less-useful auxiliary task the network will *learn* the structure of natural language. It has been argued [72], that self-supervised learning is a way to build background knowledge and of common sense in AI systems.

One of the most powerful self-supervised methods: contrastive methods [71, 26] learn an embedded space where *similar* pairs of data are mapped close to each other, and *different* pairs are mapped away from each other. A somewhat related concept is metric learning [65], where a similarly metric (This use of metric is not always consistent with the mathematical notion of a metric) is learned using data.

1.1.2.1 Deep Neural Networks: Learning & Optimization

We have used neural networks heavily in all of the work discussed in the main text. The general and optimization theory of deep neural networks are active research areas, with

[6] The distinction between self-supervised learning and unsupervised learning is often blurred.

plenty of questions yet to be answered. We will still mention some fundamental concepts of the learning problem and optimization methods briefly. For ease of notation, we will use a supervised learning setting to define and discuss the learning problem. It is, however trivial to modify this to unsupervised or semi-supervised problems.

Typically machine learning models aim to learn the parameters of a model such that the *expected* performance or cost is optimized using some criteria [44, 14].

Formally, assuming a cost function L and a supervised learning problem we can define the objective as a mapping from the parameter space as:

$$J(\theta) = \mathbb{E}_{(x,y)\sim p_{data}}[L(f_\theta(x), y)] \tag{20}$$

Where p_{data} is the data generating distribution. The hope is to find the parameters θ, such that Equation 20 is minimized. Of course, in practice, we don't know the underlying data generating distribution, but we assume we have a training set and a test set which were sampled from the data generating distribution. Then the data distribution can be estimated by the empirical distribution induced by the necessarily finite training data. Assuming we have N, (x, y) pairs of training data. We can write the empirical loss (or empirical risk) as :

$$J(\theta) = \frac{\Sigma_{i=0}^{N} L(f_\theta(x_i), y_i)}{N} \tag{21}$$

The process of minimizing Equation 21 is called empirical risk minimization. The standard methods use some variation of gradient descent based optimization algorithms. Therefore, L is assumed to be differentiable and when the desired objective is non-differentiable (For example, the accuracy of a classification task), a smooth surrogate objective is used (For classification, this is usually a form of log-likelihood).

It should be noted that the desired goal is not to compute the mathematical minimum of 21 in the parameter space, but to minimize the *generalization* error (Informally, the error on unseen data). Modern neural networks are also heavily over-parameterized and have a high capacity. Thus, in practice a number of regularization methods are also used to prevent over-fitting to the training data. In the context of neural networks the simplest of which would be early stopping, which is monitoring a corresponding metric of a validation data set, and stopping training, when the validation metric doesn't improve.

Further, given the volume of data used it is usually computationally prohibitive to optimize 21 on the whole dataset, therefore a mini-batch based optimization scheme is used: typically a variation of stochastic gradient descent (SGD). The idea is to estimate the gradients of 21 using a randomly sampled mini batch of data, usually several magnitudes smaller than the full dataset. We typically require the data in each batch to be identically, independently distributed so the mini-batch gradient is an unbiased estimator of the full gradient. Then this process is iteratively performed where the dataset is randomly divided into, mini-batches, and for each mini-batch a gradient descent step is taken with an appropriate learning rate. Typically several passes over the whole dataset are taken. It has been argued that smaller batch sizes also add a regularizing effect [44].

Optimizing deep networks is however subject to considerable challenges, and to date despite the empirical success, several theoretical questions remain open. For example, 21 is usually severely non-convex, thus there are no guarantees that the optimization algorithms will find the global minimum. It may also contain saddle points and sub-optimal local minimums. However, empirically, SGD seems to achieve satisfactory performance and the classical learning theory is deemed insufficient to explain or understand many observed phenomena of deep learning [14].

The mathematical analysis of deep neural networks is still at its infancy. We refer the reader to [14] for a discussion of the current theoretical efforts and progress.

1.2 Sepsis And Its Patho-physiology

Sepsis is a life threatening condition characterized by a pathological response to an underlying infection, resulting in severe organ and tissue damage. Sepsis has enormous mortality, morbidity and economic burden [78, 102, 91] and despite decades of research there is still significant ambiguity and even controversy regarding optimal treatment strategies [84, 53, 83]. The challenge in treating sepsis is partially caused by the heterogeneity it displays at all levels: primary infection, inflammatory response, response to treatment, and progression of the immune response.

Amongst other physiological disturbances sepsis patients exhibit display hypovolemia (abnormally low extracellular fluid), sepsis induced vasodilation (dilatation of blood vessels) vasoplegia (decreased response to compensatory mechanisms that increase vascular tone in normal physiological states). Thus, hemodynamic optimization is one of the primary goals of sepsis treatment [82]. The Surviving Sepsis Campaign [103] recommends the administration of vasopressors and fluids to counter hemodynamic abnormalities.

Vasopressors and fluids are among the treatment administrated to septic patients. We will focus exclusively on these two treatment strategies in this thesis, consistent with almost all previous work on RL for sepsis [63, 99, 74, 39]. One reason for such a focus is that there is little agreement among medical researchers on best practices that guide fluid or vasopressor administration beyond initial resuscitation. For instance, it has been shown that both vasopressors and fluids can cause negative effects in some patients [126] (For example excessive fluid administration could result in interstitial fluid accumulation and organ dysfunction).

Vasopressors are intended to counter sepsis-induced hypotension. Indeed, it has been observed that vasodilatation of systemic resistance vessels in severe sepsis can decrease by up to 75% [132]. Thus, vasopressors aim at correcting vasodilatation and vascular tone depression, as well as improving organ perfusion pressure [112]. Fluids are intended to improve tissue perfusion and oxygenation and augment cardiac output, as well as combat any hypotension caused by hypovolemia.

There are further a large variety of vasopressors and fluids at the bedside. Different vasopressors target different vascular receptors [112], among which Norepinephrine (NE) is the most commonly used. NE is also recommended as the first-line agent by the Surviving Sepsis Campaign. Other vasopressors include vasopressin, Epinephrine and Dopamine. IV fluids can be categorized as crystalloid (Solutions of ions) or colloid solutions (Suspensions of molecules in a carrier fluid) [110]. Surviving Sepsis Campaign recommends isotonic crystalloids as the first line fluid, amongst them saline (0.9% sodium chloride) being the most popular. However, there has been considerable debate on the fluid choice and the amount of fluid to be administered [110]. The differences between fluid and vasopressor types complicates the problem of mathematically modeling sepsis treatment.

Several organ function scores are used to assist clinicians and quantify organ failure. One such score is the Sequential Organ Failure Assessment or SOFA score [124]. We will use the SOFA score heavily in the sequel.

The SOFA score is based on six different scores, one for each of the respiratory, cardiovascular, liver, coagulation, renal and neurological systems. Each individual score takes an integer value from 0 to 4, inclusive of both 0 and 4: The higher the score, the worse the organ failure is. Thus SOFA score can take values from 0 to 24. A SOFA score of 2 or more is one of the necessary conditions needed to meet the current definition of sepsis.

A thorough discussion of sepsis is of course out of the scope of this text. We refer the interested reader to medical literature for more details. In addition, medical knowledge of sepsis dynamics are ever changing, and there's certainty hope that the ever-expanding research from both medical and quantitative communities will help increase this knowledge and improve the outcomes of treating sepsis.

2.0 Prologue to Article 0

The following article was originally motivated as an extended version of an oral presentation we presented at *ICCAI 2021: AI in critical illness: emergence and emergent issues*, and to be published in the Journal of Critical Care as an extended abstract. However, whilst that presentation focused on *Deep* Reinforcement Learning, we will be taking a more general view of Reinforcement Learning and Stochastic Control in this version. However, we do narrow the scope from a medical point of view, only focusing on sepsis rather than critical care applications in general.

We discuss the unique challenges critical care medicine and sepsis provide for RL based decision making systems. In each case, we present some approaches taken from different research communities to alleviate the said issues, and some promising avenues for potential solutions.

This article is intended as an introduction to the problems and promise of RL which motivated the work that follows in this thesis, as well as the bigger picture of the work. However, we have written this article in such a way that it can be read in isolation, therefore we do briefly mention some results which are explained in more detail in later articles. However, we have abridged the article to minimize repetitions. For example, the full version contains a background section, which is omitted as it is almost identical to the previous section on preliminaries. Similarly, we omitted some discussions on RL for Survival (RL4S) as this method is presented separately in Article 4.

3.0 Article 0: Reinforcement Learning & Stochastic Control for Sepsis Treatment: Challenges and Opportunities

3.1 Introduction

Recently, there has been considerable interest in learning optimal treatment strategies for septic patients directly from observational data or mathematical models [63, 28, 99, 74, 90, 92, 39, 61]. This work is motivated by a number of sound reasons: i) Sepsis has an enormous cost all around the world, in terms of mortality, morbidity and economic burden [78, 102, 91], ii) there is still ambiguity regarding optimal treatment strategies and accepted guidelines for treatment [84, 53] iii) critical care medicine is a data rich field, and the success of data driven methods in various domains suggests enormous potential positive impact if this data could be leveraged intelligently. However, there are numerous challenges at all levels when leveraging RL and optimal control theory for septic patients. Whilst some of these challenges are common to most control problems, there are plenty of challenges that are unique to critical care medicine. [1] Thus, in this article our intention is to both discuss these challenges and limitations of the current state of using computational methods to assist sepsis related clinical decision making. We further discuss relevant promising work of various research communities. However, this article is not intended as an exhaustive survey of all attempts of using control theory or RL for sepsis.

Our discussion follows a natural order. We start by discussing the stochastic control problem in the most general abstract setting [2], then move on to more standard RL and then distributional RL methods. Whilst this discussion is detailed and long, we feel it's beneficial to review the abstract problem first to provide an unified view of approaching the problem at hand. We then discuss various framing of our problem and challenges that can be faced in each, discussing appropriate solutions and methods found in the literature.

[1] We will focus on sepsis but most of these challenges are proposed solutions carry over to offline control problems in critical care medicine.

[2] As mentioned in the prologue section, this section is omitted in this text. However, we didn't modify the rest of the article for consistency.

We conclude by discussing the numerous opportunities these methods present, and promising directions for future work.

In summary, in this article :

- We identify key obstacles to RL and control applications for sepsis, and survey various solutions presented in previous work. These include sepsis specific research as well as more general work in representation learning, risk sensitive RL, uncertainty quantification and stochastic control.
- We provide a thorough discussion on quantifying the control objective and reward choice for algorithms which use some functional of cumulative sum of rewards as the objective.
- We discuss some potentially promising avenues for future research and propose novel perspectives.

3.1.1 Related Work

As interest in using RL for healthcare grew, there have been numerous reviews, and surveys on using RL for healthcare and critical care applications [77, 133]. There have also been a number of research on leveraging RL for sepsis [63, 92, 99, 90, 74] as well as guidelines for researchers [45].

The closest to this article however is [104]. The authors provide a detailed discussion on RL applications to healthcare problems in general, whilst also discussing the sepsis problem as a special case. They also discuss the challenges of defining rewards and potential solutions, which is a major focus of this work. However, in this work, we discuss a larger class of RL and control methods. Moreover, we focus exclusively on sepsis and thus the proposals and challenges are sepsis focused. We also discuss the control problem from classical stochastic control literature [15], hoping the abstract point of view will inspire novel algorithms more suited for critical care medicine. However, for the sake of completeness, we do mention some challenges and solutions that were already discussed in [104].

3.2 Background

Omitted due to the significant overlap with the previous chapter. Please refer Chapter 1.1.

3.3 Reinforcement Learning & Control for Sepsis

The general framework introduced in the previous section is naturally suited to formalize clinical decision making at the ICU. Indeed, it is well matched to the actual behavior of physicians, who observe, summarize a patient's condition, and react with the goal of maximizing chances of survival and the patient's overall health. However, there are significant challenges at all levels, and we intend to discuss these challenges and solutions proposed by different research communities: proposing some novel avenues for some of them. We will, however focus exclusively at the research level, and do not mention the considerable issues involved in the production of a real time decision support system. Nor do we discuss the potential ethical issues and philosophical dilemmas involved in *any* automated approach to healthcare.

It is important to recall that our setup will be to learn optimal treatment strategies from a fixed set of trajectories. Almost any application of RL to medicine has to be done in such an offline manner. Therefore, we do note that some of these challenges can be mitigated if we had access to an accurate simulated environment, but learning such a simulator itself will be incredibly challenging, given the complexities of the septic patients.

Further, we note that, whilst this problem shares the challenges common to *all* Offline RL problems and *all* RL to healthcare problems, we do believe sepsis and critical care medicine present unique challenges. Thus we bias our discussion strongly around treating sepsis, however, we do discuss some general solutions which can be used for these applications. (For example, we discuss some recent methods introduced in representation learning which could be directly applied).

Further, the discussion will be mainly focused on RL and discrete time methods because

this is the most common setting used in recent work. However, there has been some work using continuous time optimal control methods [28].

We will now begin our discussion with first focusing on the ambiguity on how the problem should be set up.

3.3.1 Problem Setup

Armed with the lengthy discussion of the control problem and modern Deep RL methods, let's try to frame the problem of treating sepsis at the ICU. We will focus on vasopressor and fluid treatments, and whilst there can still be ambiguity on how actions should be defined (For example: discrete or continuous?, how to combine when different vasopressors are given at the same time?) it's still less challenging than identifying the relevant states and objectives. Therefore we won't focus on defining the action space any further.

We will start by discussing the control objective and defining rewards for the standard additive RL returns.

3.3.1.1 Objective & Rewards

As mentioned previously, almost all RL methods use a cumulative future rewards (return) based objective to optimize, whether it is the expected value of the return or learning the full distribution and optimizing some risk sensitive criteria. Therefore we will first focus on these objectives, and later we will explore some idealized objectives, and possible computational methods.

Focusing on additive returns: the most natural reward choice is to define terminal rewards in terms of death or release at the end of the ICU stay (say $+1$ or -1 depending on death or release or just a negative reward for death), without using any intermediate rewards. This objective makes sense as a clinical objective as the primary goal of sepsis treatment is to decrease mortality. Indeed, there are plenty of work using RL for sepsis [63, 61, 74] which have used exclusively terminal rewards. There is also support for such a reward choice from the success of RL in other domains such as learning to play games such as Chess and Go [114]. However, it is important to note that these problems are all *online* RL problems, where

an arbitrarily large amount of data can be collected, whereas our problem belongs to the class of offline (batch) RL. If an accurate simulator of septic patients were to be developed then we believe using such terminal-only rewards could suffice, as then (simulated) online learning is possible. As we stated earlier given the immense complexities and heterogeneity in septic patients learning such a simulator would itself be a significant challenge. Therefore, at the moment RL efforts are constrained by a fixed dataset of observed trajectories, and sparse rewards choices are known to induce a high sample complexity.

Therefore we hypothesize that the terminal rewards by themselves would not suffice to learn optimal policies - at least in the current data regime. And that leads to the question of how intermediate rewards should be defined. Intermediate rewards can in theory be used to reflect short term goals of the clinicians. However such shorter term goals themselves are ambiguous and do not always correlate with reduced mortality risk [46].

Further, there is enough evidence in RL, of reasonable looking intermediate rewards resulting in undesirable behavior. Therefore, it is important that the reward choice is clinically motivated and to verify that maximizing the cumulative (discounted) future rewards is indeed a desirable goal. Alternatively one could verify that the learned policy is still optimal with the previous terminal reward scheme, however this could be mathematically challenging.

[99] use a clinically motivated intermediate reward scheme using the SOFA score and lactate to define intermediate rewards. More specifically they use rewards of the form :

$$r(s_t, s_{t+1}, a) = C_0 \mathbb{I}_{\{\text{SOFA}_{t+1}=\text{SOFA}_t \ \& \ \text{SOFA}_t>0\}} + C_1(\text{SOFA}_{t+1} - \text{SOFA}_t) + C_2 \tanh(\text{Lac}_{t+1} - \text{Lac}_t)$$

Where SOFA and Lac denotes the SOFA score and lactate. A modification of this rewards was used in more recent work [90].

In addition to this reward scheme, the only other intermediate reward design we are aware of is [92]. The authors define intermediate rewards using a predictive model trained to predict probability of mortality at a given state. Then, intermediate rewards were defined in terms of log odds.

In the context of critical care applications, a well suited intermediate reward would be one of the form $R(s) - R(s')$, where R is a suitable notion of mortality risk. Then ignoring discounting, the cumulative rewards has the desirable property of minimizing the

cumulative mortality risk. Recent work [89] learns such a non-probabilistic risk score using a semi-supervised learning scheme. In particular, they project EHR data to a lower dimensional unit ball such that patients with similar mortality risk will be on the same level sphere. Then, they experiment by using the difference of the squared norm of the consecutive embedded vectors as intermediate rewards.

Another potential approach may be to use Inverse Reinforcement Learning (IRL) to learn a reward function under which the observed behavior is optimal [4]. However, the later assumption arguably makes it less attractive as by definition, the observed behavior is optimal with respect to the learned reward function. In addition, IRL comes with its own set of obstacles [4].

Even after deciding on a suitable reward choice, there is still ambiguity on which operator on the value distribution should be optimized. Distributional RL methods [11, 9, 33] are well suited to optimize a risk sensitive criteria such as C-var. The use of these methods in the context of sepsis treatment has been limited [39].

3.3.1.2 Partial Observability: State Representation

Partial observability of the state is another key challenge in applying RL techniques for critical care medicine. Recall that in a MDP formalism, the state should summarize the entire system at a given time. For septic patients, this should ideally summarize all relevant physiologic processes, diagnoses, previous treatment amongst others. Ideally, the state should be at least as rich as all the patient level information a clinician would consider before making a decision. However, as we have mentioned multiple times, despite the vast amount of data collected at the ICU, the true physiologic *state* of a patient is rarely captured by the observable data, and the complexities and the heterogeneity of septic patients make it extremely challenging to have a well defined but easily computable state.

However, there have been promising work and significant progress made by various research communities which can be readily applied to improve the temporal state representation of a patient. We will discuss some such approaches next.

Some RL focused design choices include clustering patient readouts [63], using recurrent

autoencoders to summarize the history of the physiologic time series (labs, vitals and scores) [92, 90] and using probabilistic belief methods [74]. Further, [61] empirically evaluated a number of different representation learning methods, including neural ordinary differential equations [25] and other more recent developments in Deep Learning. [76] uses a discrete state space which maximizes correlation with outcomes and interventions.

In our methods, we believe strongly in the integration of first principle based mechanistic models whenever possible. For example, in [90] we integrated a simple cardiovascular model which relates unobservable cardiovascular states to observable states, with a deep GRU based autoencoder. Then using this neural network architecture, we estimated unobservable cardiovascular states, in a patient specific manner. Augmenting the state with such clinically relevant cardiovascular states is a way to encode medical and physiological knowledge and also improves the trustworthiness of an AI system and can be used to check if the recommended policies are consistent with clinical knowledge. However, integrating such models may require more granular data than what is available in publicly available data sources.

There are also several non-RL specific representation learning methods which could benefit RL. Representation learning methods have had incredible success in various other domains such as computer vision and natural language processing [26, 48, 35]. Recently, there has been a number of work that use Deep Representation Learning for clinical time series and other biological data [113, 131, 89]. These methods could work in self-supervised, semi-supervised or even supervised settings. The promise of these methods is that they can potentially learn intelligent representations with no or minimal supervision. Even in the semi-supervised or supervised setting, the learned representations can uncover insights beyond the supervision signals provided. With proper care these methods can also be adapted to encode prior knowledge using relevant geometric priors or by modifying the optimization process.

Another benefit of representation learning methods is that they provide an unified way to encode information from multiple sources and modalities corresponding to a patient's stay and clinical history. These may include clinicians' notes and diagnoses, medical images, audio signals and possibly bio-markers, genomics amongst other modalities. In doing so the strengths of modern machine learning methods for unstructured data (especially images and natural language) can be exploited. There have been work in medical AI, which have used

multiple data modalities [119, 94, 120, 59]. For example, [94] used audio signals of cough sounds, and clinical reports to identify respiratory disorders in children. [59] used NLP to extract cancer outcomes from radiology reports. To the best of our knowledge, multi-modal models have been under-explored in RL applications, however, we do expect to see such methods in the future.

3.3.1.3 Uncertainty Quantification

Ignoring uncertainty and risk when making clinical decisions can result in catastrophic results. Further, the problem of treating sepsis is full of uncertainties at all levels. Thus, it should be of no surprise that identifying and quantifying all forms of uncertainties are both incredibly challenging but extremely important. Indeed, it has been argued that a principled and a formal approach to uncertainty quantification (UQ) is essential for *any* machine assisted clinical decision making system [7, 64]. A full discussion of all possible dimensions of uncertainty however is not possible within the scope of this text (But we have already discussed some forms of uncertainty, for example of the state in the previous section), so we will focus on some prominent aspects of UQ, and how they affect the sepsis control problem.

Uncertainty is usually classified into two broad categories. Aleatoric (environment) uncertainty and epistemic (model, parametric) uncertainty [52]. Although the definitions of these terms could be nebulous and inconsistent among different sources, at a high level aleatoric uncertainty denotes the inherent and irreducible uncertainty within a system of interest. In contrast, epistemic uncertainty denotes the uncertainty resulting from a lack of knowledge. This itself is a broad definition and includes a variety of forms of uncertainty. For example, when a model is used, epistemic uncertainty include: the uncertainty of the model on its own outputs, model limitations and (if the model is parametric) the uncertainty of its true parameters. In offline RL, a clear form of epistemic uncertainty is caused by the quantity and the quality of data available. Quantifying and acknowledging epistemic uncertainty is in particular important in any application of deep RL, because it is well known that deep neural networks can produce unreliable outputs when the inputs are away from the distribution it

was trained on.

Distributional RL methods are well suited to quantify the inherent environment (aleatoric) uncertainty over future rewards [3] [11, 33, 9]. In other words, these methods acknowledge that due to the randomness of the environment dynamics and possibly the reward process itself, the return is a random variable and its full distribution naturally gives more information than a scalar quantity such as an expected value or a percentile. The risk sensitive operators on the return distribution and risk sensitive distributional RL methods which we previously discussed attempt to minimize adverse results from aleatoric uncertainty [4]. Therefore we believe that distributional methods are well suited for RL applications for medicine.

Quantifying epistemic uncertainty could be more complicated and depends on the algorithms used. Bayesian methods are a natural fit for most UQ problems. and there are plenty of work which use Bayesian methods for optimal control [101, 30, 5]. For deep learning based systems (not necessarily in the context of RL) [22] identifies three most common UQ methods: Bayesian Neural Networks (BNN), Concrete Dropout (CD), and Deep Ensembles (DE). Any of these models can be used whenever a deep neural network is used in the RL setting (for example to represent a policy, a value function or environment dynamics).

There is work that integrate UQ with RL for sepsis. The main motivations of these methods include: quantifying the confidence of each proposed action and potentially abstaining from recommending an action if and when the uncertainty is too high for a given patient state, preventing the recommendation of any dangerous or risky strategies and quantifying the effects of distributional shift. [90] uses a distributional RL agent to learn the environment uncertainty and quantify epistemic uncertainty using deep bootstrapped ensembles. They also propose a preference score to recommend decisions after discounting high model uncertainties (And also considering an estimated probability). This preference score can directly be used in the optimization (ignoring possible computational resource constraints), in an Actor-Critic framework [116]. [39] decomposes the two types of uncertainties. Essentially, they take a Bayesian approach to learn the quantiles of the return distribution. [76] uses a method where

[3] This however is not the only form of environment uncertainty, for example, physiological processes and treatment responses have inherent randomness, however at least in a model free RL setting these can be captured by the randomness of the returns.

[4] Although some such methods can be adapted to factor in model uncertainty.

they compute confidence bounds on Q values of sepsis interventions. Outside of medicine, there have been various uncertainty aware methods for RL in safety critical domains [58, 80].

Before concluding this section, we again emphasize that UQ is one of most important components of any clinical decision making system and refer the reader to UQ focused articles for a more thorough treatment. [7, 64, 111, 1, 22]

3.3.2 Deep Reinforcement Learning & Algorithmic Challenges

Most modern success of RL has resulted from using deep neural networks as function approximators. However deep learning and in particular deep reinforcement learning do not yet have the strong theoretical guarantees that more traditional methods possess.

In online environments, this may not be a major obstacle as the performance of a policy can always be verified by interacting with the environment multiple times. However in offline RL, especially in critical care medicine the lack of clear evaluation criteria (especially when the rewards themselves are ambiguous), makes deciding between different model classes, hyper-parameter choices and even detecting over-fitting extremely challenging.

We note that given the complex dynamics of septic patients and the high dimensional and possible multi-modal nature of the patient's state, deep learning based methods are better suited as function approximators than most alternatives. However, it is important to be aware of the limitations and epistemic uncertainties of the models. A reasonable strategy would be to use UQ to incorporate a confidence level with each decision and recommend an approximate behavior policy if the confidence is low. This approach is also recommended in [64] and [46].

3.3.3 Explainability & Trustworthiness

The ability to explain its decisions is a highly desirable feature in any computational medical decision support system. Unfortunately, almost all machine learning methods and even traditional engineering and control methods can fail to sufficiently explain their choices.

In the context of modern RL and AI, explainable artificial intelligence (XAI) is an active and rapidly progressing research area, but most of the research has focused on supervised

learning. [98] [128] and [49] provide comprehensive surveys of recent efforts to make RL more explainable and trustworthy. [128] divide these efforts into either: providing query-based explanations, summarizing learned policies, human-in-the-loop collaboration, verification of systems using expert knowledge, or highlighting visualizations.

For treating sepsis, explaining each individual treatment recommendation would be very challenging, and to the best of our knowledge, the current state of RL does not include a well accepted procedure to provide such an explanation. However, as we have mentioned previously, one way to provide *some* explainability and improve trustworthiness is to encode medical knowledge using physiological models. Whilst this would not make the system explain every individual decision made, it can provide some form of explainability by query based explanations and policy summarizing (by comparing the recommended treatment with relevant physiological states). For example, vasopressors are intended to increase systemic vascular resistance (SVR) therefore if the system on average, associates more frequent vasopressor recommendations with decreased SVR, the confidence of the system would improve.

We also strongly believe in having human experts in the loop whenever possible, from the design of systems to evaluating and eventual deployment. Amongst other things, clinicians would be best equipped to evaluate decisions made by a RL system. All automated decision making systems can only be used to supplement and support clinicians, at least in the foreseeable future. Therefore, having clinicians involved from the start could significantly accelerate progress.

Further, it is important to be aware of the limitations of RL systems. For example, uncertainty quantification discussed previously, would increase the trustworthiness of an AI system. Other factors to address include the validation and transparency of the data the model was trained on, identification and elimination of any opportunities that negative human biases could have been encoded, robustness of the policies (For example, is there any possibility of adversarial attacks?) and verification of all assumptions.

3.3.4 Evaluation

The lack of a well defined criteria to evaluate the performance of a learned policy is arguably the biggest challenge faced by RL applications to medicine. However this aspect has been discussed in detail in other work [46, 104, 45], and thus we will keep our presentation brief.

Currently, the best (and arguably the only) possible solution seems to be off-policy evaluation (OPE) [118, 117, 97], using a validation dataset. Informally, OPE attempts to estimate the value (expected cumulative discounted future rewards) or some other functional of a policy using data generated by another behavior policy. These efforts can be divided into three methods, importance sampling based methods, model based methods or a combination of the two approaches. However, the most widely used and unbiased OPE estimators are importance sampling based estimators.

It has been argued that *all* OPE methods are ill-suited for clinical applications [46]. Indeed, arbitrarily bad policies can result in high OPE estimates. For example, consider importance sampling based methods where the return of each trajectory is re-weighted using a product of importance ratios $\prod_{t=0}^{T} \frac{\pi_e(a_t|s_t)}{\pi_b(a_t|s_t)}$ where π_e and π_b are the probability of the action a_t taken at state s_t in the policy we are evaluating and the behavior policy respectively [5], and T is the trajectory length. Notice that a_t is the action taken in the observed dataset. Now, suppose a RL agent recommends not to administer any treatment to patients with high mortality risk. Therefore for most non-survivors, the importance ratio would approach zero, as it is very likely that the observed data, generated by human clinicians would have administered some treatment for these patients and for those actions the evaluating policy will have 0 probability. Of course, these non-survivors are the source of almost all negative rewards, regardless of the reward formulation. Thus, our hypothetical RL agent would have eliminated all possible negative rewards. Another issue is, as mentioned in [46], if the evaluated policy is deterministic, the importance weights of a trajectory will be identically zero unless the actions match with the observed actions at *all* time steps.

Further, by definition OPE estimates are defined in terms of the rewards. Therefore, all

[5] For simplicity we have assumed that the action space is discrete, if not and the densities exist, the ratios can be defined in terms of densities

the obstacles in identifying rewards would carry over.

A more heuristic approach to evaluating policies, used in healthcare literature is to compare the difference between the observed actions and the recommended actions with an outcome such as mortality [46]. However [46] shows how these methods are prone to confounding factors.

3.4 Opportunities and Directions for Future Research

Our presentation so far was focused on the challenges of leveraging RL and control to learn optimal treatment strategies, however, its potential cannot be understated or undermined. Whilst we certainly don't endorse any notion of replacing human clinicians, RL based systems can be and arguably already is a valuable clinical decision support tool. As an example a RL system can recommend a policy, its confidence in the policy and an approximate behavior policy using imitation learning (as in for example [90]). Then, the clinician can use their own judgment and decide on the appropriate treatment. Even ignoring the chance of RL uncovering potentially *better* strategies, an accurate imitation learner itself can provide valuable information, for inexperienced clinicians. There are plenty of situations across the world where human expertise is sparse [6], and an imitation learner can provide an approximate action using data generated by thousands of human clinicians. This view is also echoed in [46], where the authors claim "Retrospective critical care data sets such as the one we described are a gold mine of information, and to dismiss them entirely would be the equivalent of telling a clinician to avoid learning from their colleagues and focus only on their own experience. Just as it is unethical to present sloppy research results, it is also unethical to not leverage data that could improve patient health".

Another realistic potential of these methods is to uncover hints towards novel treatment strategies, which of course should be subject to clinical verification. For example recent work has suggested hints towards vasopressor policies [90] which were consistent with recent

[6]However, it is important that such a system is trained and verified on data of that particular region, the current ICU data is almost exclusively from North America or Europe.

medical literature [112].

The challenges discussed previously provides an opportunity to develop sophisticated control methods that are specifically suited for critical care medicine. To develop such methods, we strongly believe in an inter-disciplinary approach. As mentioned earlier even from a pure computational and mathematical perspective control theory was historically dispersed across various disciplines, each focusing on a specific case that is only slightly different from the general problem. However, there are recent attempts to unify these methodologies into a common framework. A potential benefit of these attempts among others would be establishing stronger theoretical results for (Deep) RL problems. Indeed, this was a motivation for describing the abstract control problem in the background section.

From a medical perspective, it's imperative to have clinicians, medical scientists and physiologists involved. There is a large range of opportunities to encode domain knowledge which could mitigate some of the challenges described earlier. These include, defining and potentially augmenting the state representation, regularizing the policies to ensure safety, defining new algorithms and using model based methods. For the later case, a potentially fruitful approach is to incorporate first principle based mechanistic models. However, most physiological models are defined in continuous time using differential equations, and may only hold for a short time span. Therefore, it is most immediately clear how these models can be used in a discrete time control problem. However, a continuous time stochastic control problem for septic treatment has been proposed in [28]. Continuous time methods are certainly under-explored and could be an important direction to follow in the future. However, the unknown dynamics and the complexities of the sepsis problem some physiologic models themselves, (which as all models are approximations) may not be appropriate for septic patients, therefore care should be taken.

However, we do believe there are a lot of opportunities to use methods from traditional mathematics to improve the current status quo. In addition to physiological modeling, these include uncertainty quantification, causal inference, differential geometry (for geometric deep representation learning methods [19]) and stochastic analysis. Of course any potential AI application to healthcare would have numerous ethical, sociological and educational implications which were not considered in the scope of this text. We do note that even in

the computational perspective taken here, some of these dimensions cannot be ignored. For example, most current work using RL for sepsis uses the MIMIC-III database. Whilst, this is a rich data source, the patients are only from two intensive care units located in Boston. Therefore there is an obvious lack of representation of critically ill patients from other parts of the world, whose treatment responses could be different due to a number of reasons. Thus, care should be taken when interpreting the results of any RL system when a potential new patient is away from the training data distribution.

4.0 Prologue to Article 1

The following article titled *Unifying Cardiovascular Modelling with Deep Reinforcement Learning for Uncertainty Aware Control of Sepsis Treatment* is published in **PLOS Digital Health** [90]. What follows is presented in the exact same way as the published article, followed by its supplementary information. Note that, due to formatting conventions the methods are presented after the results.

This work addresses two main challenges in Reinforcement Learning applications for sepsis: Partial Observability and Uncertainty.

In particular, we leverage mechanistic mathematical models which embody decades of medical research to introduce a novel *physiology aware* neural network architecture. This network is trained in an unsupervised manner, to dynamically estimate personalized unobservable cardiovascular states. Augmenting the state with the learned cardiovascular representation and another recurrent neural network based representation for lab history, we use Deep Distributional Reinforcement Learning to learn value distributions. We further, mathematically define parametric uncertainty for Offline RL, and quantify the uncertainty of the results. Moreover, we introduce a framework for uncertainty-aware decision support with humans in the loop.

We show that our method learns physiologically explainable, robust policies, that are consistent with clinical knowledge. Further, our method consistently identifies high-risk states that lead to death, which could *potentially* benefit from more frequent vasopressor administration, providing valuable guidance for future research.

This work is co-authored by committee members Dr. Christopher James Langmead, Dr. Gilles Clermont, and Dr. David Swigon.

5.0 Article 1: Unifying Cardiovascular Modelling with Deep Reinforcement Learning for Uncertainty Aware Control of Sepsis Treatment

Sepsis is a major host response to infection which can result in tissue damage, organ damage and death. The mortality and economic burden of sepsis is very large. In the U.S., sepsis is responsible for 6% of all hospitalizations and 35% of all in-hospital deaths [78, 102], and an economic burden of more than $20B per year [91]. The treatment of sepsis is extremely challenging, due to the high variability among patients, with respect to both the progression of the disease, the host response to infection, and the response to medical interventions, suggesting the need for a dynamic and personalized approach to treatment [84, 70, 36]. Presently, the search for treatment strategies to optimize sepsis patient outcomes remains an open challenge in critical care medicine, despite decades of research.

Recently, there has been considerable interest in the application of Reinforcement Learning (RL) [116] to extract vasopressor and intravenous (IV) fluid treatment policies (i.e., strategies) for septic patients from electronic health records data (ex. [63, 99, 92, 74, 61]). Informally, the goal is to learn a policy that maps the patient's current state to an action (i.e., medical intervention), so as to maximize the chances of future recovery. The RL framework is well-matched to the actual behaviors of physicians, who continuously observe, interpret, and react to their patient's condition. The promise of RL in medicine is that we *might* be able to find policies that outperform humans (as it has in other domains, ex. [87, 114, 41]), by automatically personalizing the treatment strategy for each patient, as opposed to using one that is expected to work well on the *typical* patient [77, 133]. However, there are many challenges that must be met before RL can be used to guide medical decision making in real-life settings [45].

A particularly severe challenge is partial observability of patient state. Despite the richness of data collected at the ICU, the mapping between true patient states and clinical observables is often ambiguous. We believe that this ambiguity can be reduced through the use of mechanistic mathematical models of physiology that relate observables to a more complete representation of the patient's cardiovascular state. Such models are plentiful in

the literature, and embody decades of research in physiology and medicine. Our proposed solution integrates, for the first time, a clinically relevant mechanistic model into a Deep RL framework. The specific model we use was chosen because it estimates the unobservable aspects of cardiovascular state that are relevant to specific interventions (vasopressors and IV fluids), and the clinician's goals — counteracting hypovolemia, vasodilation, and other physiological disturbances. This model is integrated into our framework using a self-trained deep recurrent autoencoder that uses a variety of inputs, including the patient's vital signs, organ function scores, and previous treatments.

The second challenge addressed by our framework is uncertainty in the learned policy, and thus the expected outcomes. Similar to previous efforts to extract sepsis treatment policies from retrospective data (ex. [63]), our method works in the Batch Reinforcement Learning setting [68], where the agent cannot explore the environment freely. In this setting, it is well known that RL can perform poorly [42], if the agent encounters states that are rare or even unobserved in the training data. For this reason, it has been argued that all forms of uncertainty should be quantified in any application of Artificial Intelligence to Medicine [8]. Thus, we quantify model uncertainty[1] via bootstrapping and take a distributional approach to factor in environment uncertainty. We also propose a decision framework where the clinician is presented with a quantitative assessment of the distribution over outcomes for each state-action pair.

5.1 Background & Related work

5.1.1 Reinforcement Learning

Reinforcement Learning is a framework for optimizing sequential decision making. In its standard form, a Markov Decision Process (MDP), consisting of a 5-tuple (S,A,r,γ,p) is the framework considered. Here, S and A are state and action spaces, $r : (S, A, S) \to \mathbb{R}$

[1]This should not be confused with the model-based vs model-free RL distinction, because once we have inferred latent states, our approach qualifies as 'model-free'. The literature also uses the term epistemic uncertainty and parametric uncertainty for model uncertainty.

is a reward function, $p : (S, A, S) \to [0, \infty)$ denotes the unknown environment dynamics, which specifies the distribution of the next state s', given the state-action pair (s, a), and γ is a discount rate applied to rewards. A policy is (a possibly stochastic) mapping from S to A. The agent aims to compute the policy π which maximizes the expected future reward $E_{p,\pi}[\Sigma_t \gamma^t r_t]$. In the partially observed setting there is a distinction between the observations, denoted as o_t, and the state s_t, and the environment dynamics includes the conditional probability density $p(o_t|s_t)$. This extends the MDP formalism to that of Partially Observed Markov Decision Process (POMDP).

The search for of an optimal policy can be performed in several ways, including the iterative calculation of the *value function*, $V^\pi(s) = \mathbb{E}_{p,\pi}[\Sigma_t \gamma^t r_t(s_t, a_t)|s_0 = s, \pi], \forall s \in S$, which returns the expected future discounted rewards when following policy π and starting from the state s, or the *Q-function*, $Q^\pi(s, a) = \mathbb{E}_{p,\pi}[\Sigma_t \gamma^t r_t(s_t, a_t)|s_0 = s, \pi, a_0 = a], \forall s \in S, a \in A$, which returns the expected future reward when choosing action a in state s, and then following policy π. Central to many RL algorithms is the Bellman equation [12]:

$$Q^\pi(s, a) = \mathbb{E}_p[r(s, a)] + \gamma \mathbb{E}_{p,\pi}[Q^\pi(s', a')], \tag{22}$$

and the Bellman optimality equation:

$$Q^*(s, a) = \mathbb{E}_p[r(s, a)] + \gamma \mathbb{E}_p[\max_{a' \in A} Q^*(s', a')] \tag{23}$$

(where $Q^*(s, a)$ is the optimal Q function, and s' denotes the random next state).

5.1.2 Distributional & Uncertainty Aware Reinforcement Learning

Distributional Reinforcement Learning [10, 107, 6] extends traditional RL methods by estimating the entire return distribution from a given state, rather than simply an expected value. It has been shown that distributional RL can achieve superior performance in the context of Batch RL [2]. For this reason, and because distributions are relevant to our overall goal of providing clinicians with an assessment of the range of possible outcomes for each state-action pair, we employ Categorical Distributional RL [10]. Here the state, action value distribution is approximated by a discrete distribution with equally spaced support. Further,

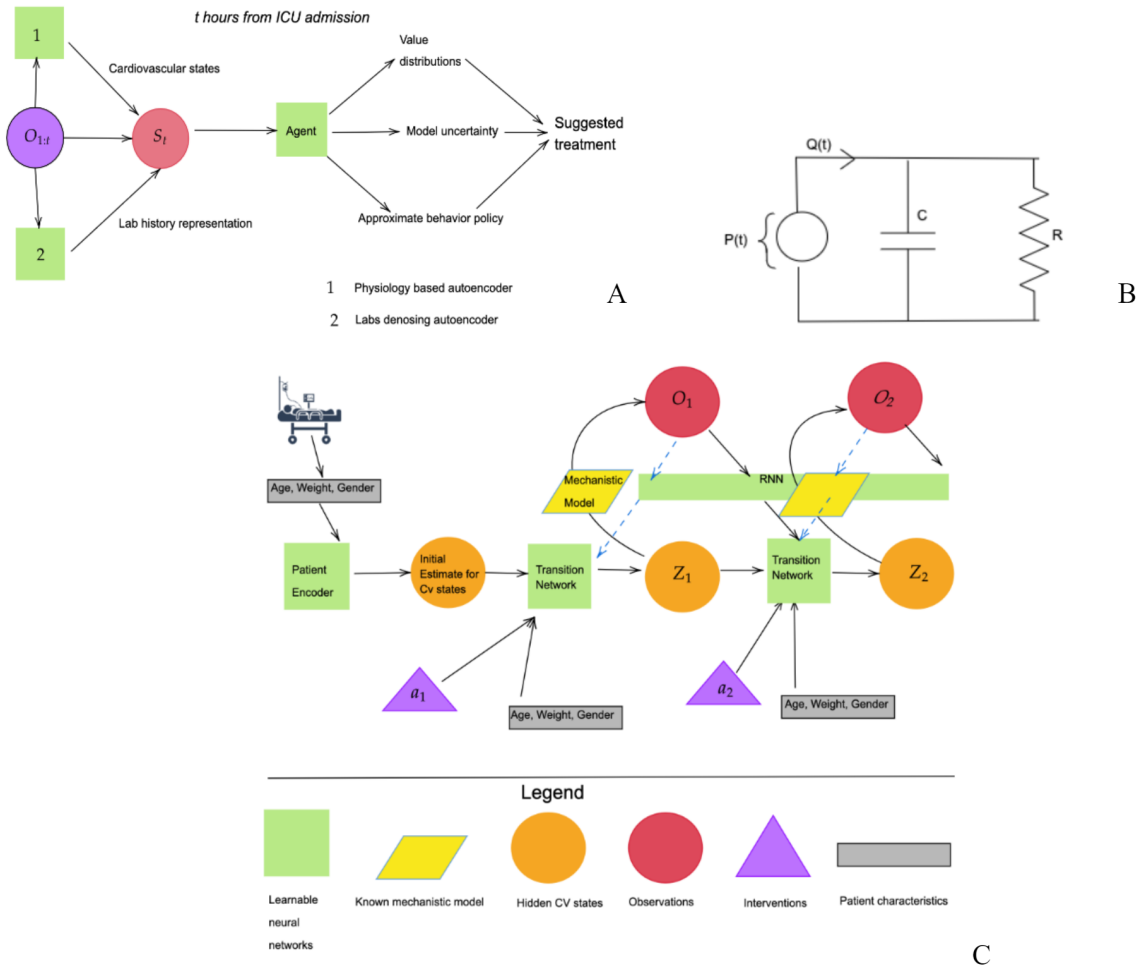


Fig. 1: **Proposed decision support system (A)**: We use the compete patient history, which includes, vitals, scores, and labs, and previous treatment, to infer hidden states. These would all combine to make the state S_t. Our trained agent, takes this state and outputs value distributions for each treatment, its own uncertainty, and an approximate clinician's policy. We then factor in all 3 to propose uncertainty-aware treatment strategies. **The electrical analog of the cardiovascular model (B)** This provides a lumped representation of the resistive and elastic properties of the entire arterial circulation using just two elements, a resistance R and a capacitance C. This model is used to derive algebraic equations relating R, C, stroke volume (SV), filling time (T), to heart rate (F) and pressure. The Cardiac Output (CO) can be then computed as (SV)F. These equations define the decoder of the physiology-driven autoencoder. **Complete physiology-driven autoencoder network structure (C)** Patient history is sequentially encoded using three neural networks. A patient encoder computes initial cardiovascular state estimates using patient characteristics, a recurrent neural network (RNN) encodes the past history of vitals and scores, up to and including the current time point, and a transition network which takes the previous cardiovascular state, the action and the history representation to output new cardiovascular state estimates.

we employ Deep Ensembles [22] to quantify the uncertainty associated with each state action pair. These ensembles are constructed using bootstrap estimates, as explained in the methods section.

5.1.3 Reinforcement Learning in Medicine

Reinforcement Learning has been used for various healthcare applications. References [133] and [77] provide comprehensive surveys of healthcare and critical care applications respectively. In the specific context of sepsis treatment, Komorowski *et al.* [63] used a discrete state representation created by clustering patient physiological readouts, and a 25 dimensional discrete action space to compute optimal treatment strategies using dynamic programming based methods. Others have considered continuous state representations [99] and partial observability [92].

Our proposed decision support system is based on a preference score as shown in Fig 1A. In contrast to previous work, we choose a lower dimensional action space (9 actions), to ensure sufficient coverage in the training data, and a reduced decision time-scale, to be more aligned with clinical practice. The short time scale also provides a clinical justification for the less granular action space. Our rewards are based on previous work [99] (see Methods), which has intermediate SOFA-based rewards, and ± 15 terminal rewards, depending on survival.

5.2 Results

5.2.1 Trajectory Reconstruction Using the Physiology-driven Autoencoder

One of the key features of our method is the physiology-driven structure of the autoencoder that represents the cardiovascular state of the patient (see Fig 1B and Fig 1C). The decoder of this autoencoder is a set of algebraic equations that map the latent state to observable, and clinically relevant physiological parameters, such as heart rate and blood pressure. Fig 2 shows selected reconstructed trajectories for one representative patient, using various levels of data corruption (see Methods). As the figure illustrates, the model successfully reconstructs

the observable outputs and their trends with corruption probabilities as high as 25%. It is only at extreme levels of corruption (50%) that the model's accuracy degrades. Such robustness to moderate levels of corruption was typical among training and validation patient trajectories. We thus conclude that the autoencoder has learned an effective representation of the cardiovascular state of the patient.

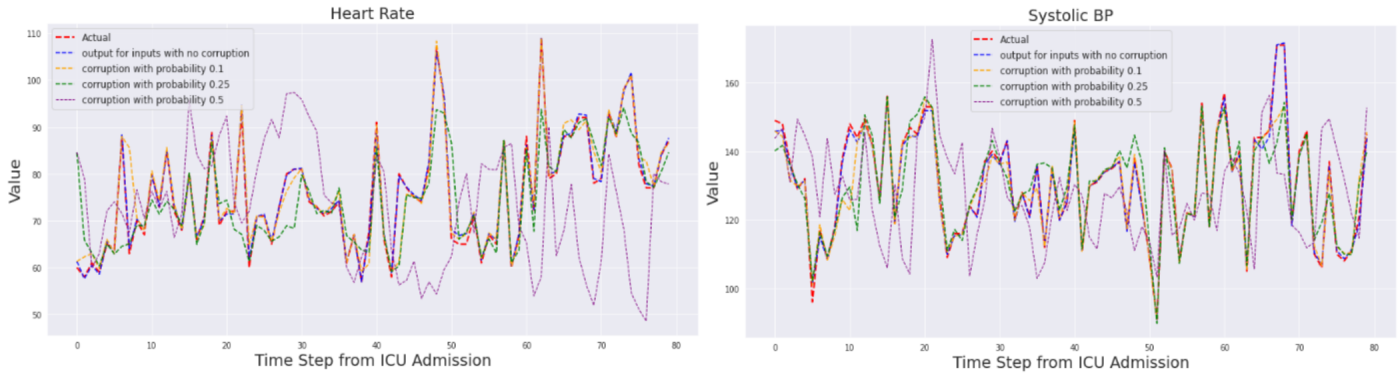


Fig. 2: Reconstruction of two validation patient trajectories using different levels of corruption using the physiology-driven autoencoder, **Left:** Heart Rate. **Right:** Systolic Blood Pressure.

Below, we present (in Table 1) average unnormalized mean square error of the four dimensional output, per time step to the nearest integer.

Table 1: Mean square error of reconstruction

Corruption probability	MSE per time step
0%	6
10%	45
25%	59
50%	258

5.2.2 Value Distributions & Expected Values

We next investigated whether the learned values are generalizable, consistent with clinical knowledge, and correlated with the risk of death in non-survivors. To do this, we examined the value distributions that are produced at each time-step for patients in the validation

set, stratified by outcome (i.e., survivor vs non-survivor). Fig 3 plots the average value distributions output for non-survivors (top) and survivors (bottom) at 48, 24, and 1 hour from death or discharge. The individual lines in each panel correspond to the value distributions under the nine discrete actions available to the agent. We emphasize that these plots were generated for the purpose of analyzing the learned models. In particular, the network only sees the current state when it outputs such distributions; it is not given with any information about the future.

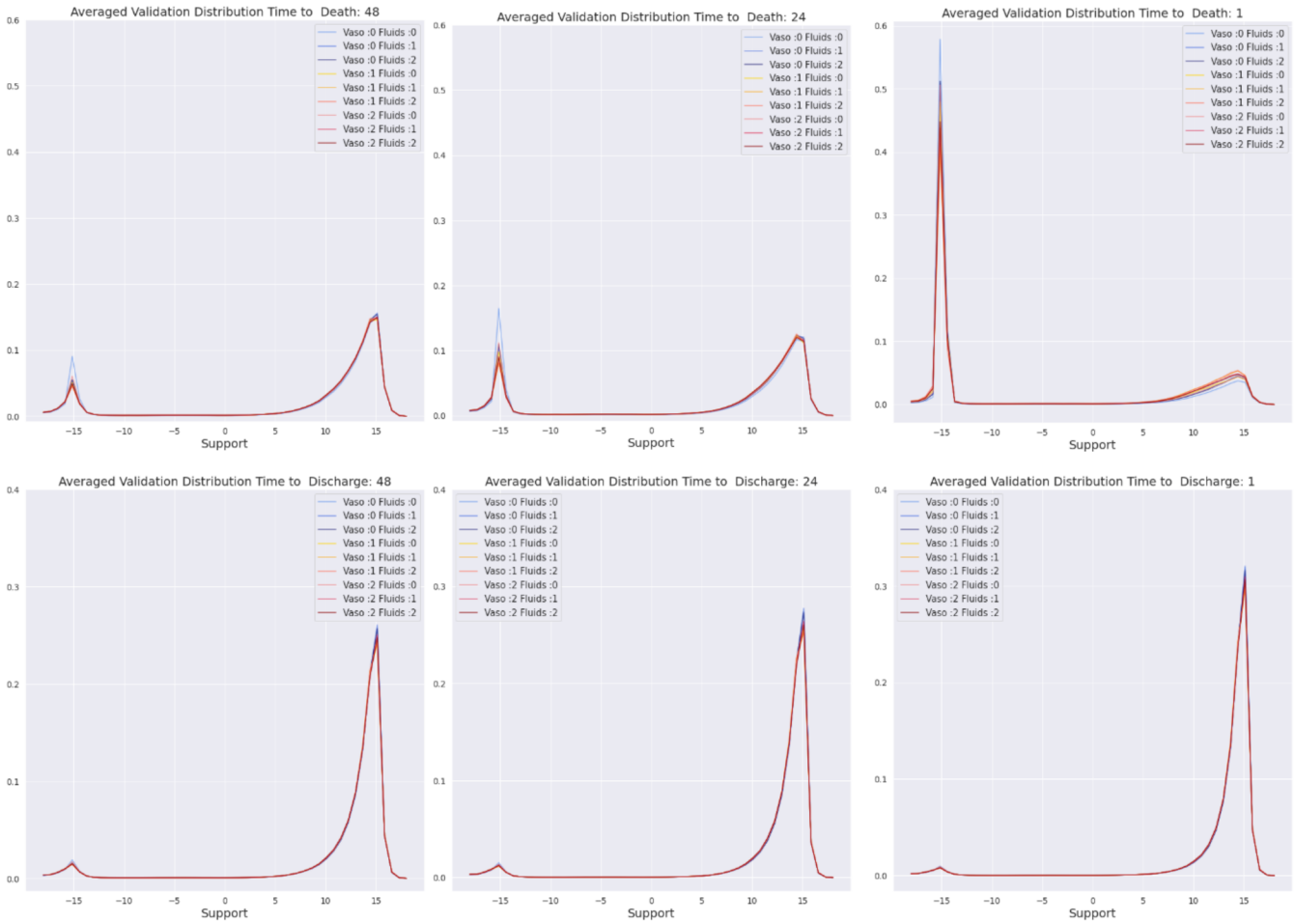
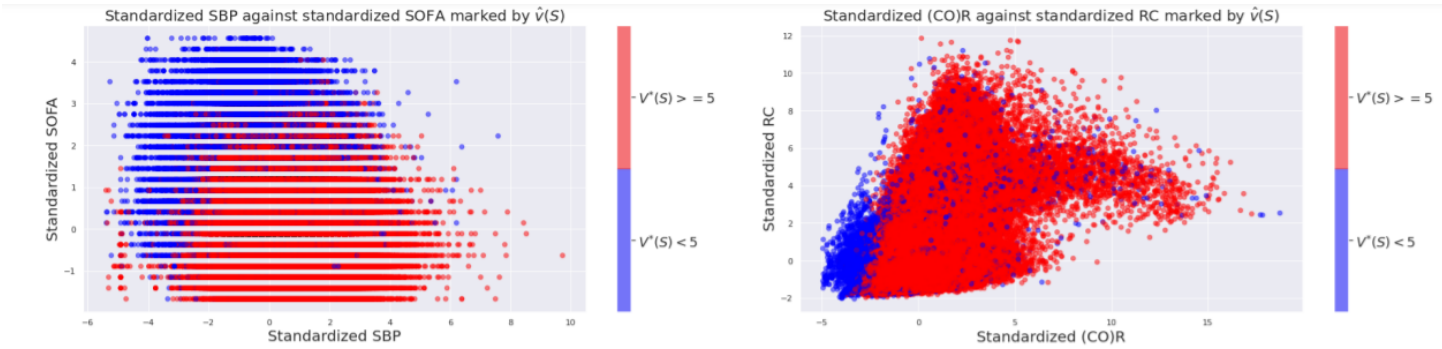


Fig. 3: Value distributions for validation patients averaged according to different times from death or discharge, **Top row**: Non Survivors. **Bottom row**: Survivors.

Fig 3 clearly exhibits bi-modal distributions over values for non-survivors as much as 48 hours in advance of death. Further, as the patient gets closer to death, the mass shifts



Feature	Importance Score
CNS	0.664190
BUN	0.462304
SOFA	0.418309
HR	0.362906
HB	0.264061
l_3	0.253673
R	0.226833
SV	0.224609
MBP	0.224562
AG	0.214475

Fig. 4: Scatter plots of scaled features : **Top row**: Marker colors indicates if $\hat{V}^*(S) < 5$ (Blue) or $\hat{V}^*(S) \geq 5$ (Red) **Bottom**: Top 10 features measured by feature permutation. Here, l_k denotes the kth component of the latent lab representation.

towards the left peak (which corresponds to death). This behavior is consistent with the patient's deteriorating condition. Additionally, the distribution associated with the "no treatment" action has a larger left peak than others, highlighting that for these states the lack of treatment for even one hour can be fatal. The mass of the distributions for survivors, in contrast, is concentrated closer to the right limit *and* there is little difference between actions. Both of these observations are consistent with the expectation that survivors are less likely than non-survivors to enter the highest risk states, and so the consequences of a change in action/treatment are less extreme.

We then investigated the dependence of features and inferred states on the value distribu-

tions and determined that they are explainable, and consistent with clinical expectation. For example, Fig 4 shows two scatter plots contrasting representative pairs of variables, stratified by an optimal expected value threshold of five. (This threshold was chosen arbitrarily, and we could observe similar results for any reasonable threshold.) It is clear that the model associates different states with different expected rewards/risk. For example, the model associates low SBP (hypotension) and high SOFA scores with an increased risk of death, which is consistent with medical knowledge. Thus the agent has learned to discriminate between low and high risk states in an explainable manner. The ability to learn such associations is noteworthy because the training and test data are highly imbalanced. In particular, 89% of states have the property $V^* \geq 5$.

Finally, we quantified the importance of each feature using feature permutation [88]. Briefly, for each patient we permute a selected feature while keeping others fixed. The mean absolute value difference of the Q function (across states and actions) is taken as the importance score for that patient. The above table lists the top 5 features across the entire cohort. The complete feature ranking can be found in the supplementary materials (Appendix C in S1 text). The cardiovascular states and the latent lab representations are among the most important features, highlighting the importance of representation learning.

5.2.3 Vasopressor Treatment Strategies

We observed that the RL agents consistently recommend vasopressors for near-death (non-survivor) states, and that the percentage of such states increase closer to the patient's eventual death. This phenomena is also shared by validation cohort states, as illustrated in Fig 5A, suggesting that this behavior isn't due to overfitting. In contrast, clinicians have only administered vasopressors on average around 40% of the time, and this number drops off rapidly in the last 10 hours. We investigated whether these differences are an artifact of our choice of method by evaluating different training options and algorithms. Specifically, we: (i) trained networks with and without weighted experience sampling scheme (explained under Methods); (ii) used a different distributional RL algorithm, called Quartile Regression Q Learning [33]; (iii) considered an artificial voting ensemble agent, which only administers

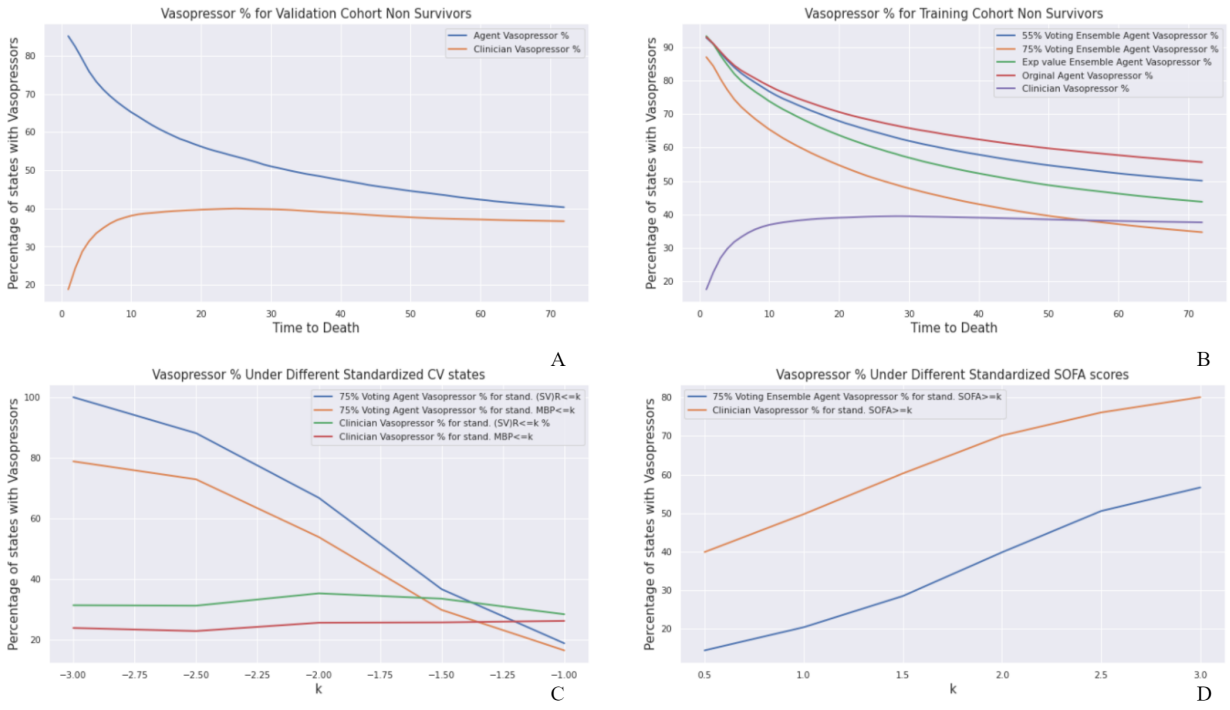


Fig. 5: **Top row**: Percentage of states with vasopressors recommended for the training and validation states, with time to eventual death. Here a $p\%$ voting agent, denotes an agent which only prescribes vasopressors if an only if least $p\%$ of the Bootstrapped Ensembles have agree on giving vasopressors. **Bottom row**: The percentages of states with vasopressors recommended or given with respect to cardiovascular states and SOFA score.

vasopressors if at least $p\%$ of the ensemble agrees on giving vasopressors, at a given state; and (iv) consider the expected value of the ensemble agent, which takes a weighted average (weighted by the number of patients it's trained on) of expected values of each bootstrapped network. In each case we observed similar results, as shown in Fig 5B.

We also investigated the relationship between vasopressor recommendation and cardiovascular states, and SOFA score. As illustrated in Fig 5C, the RL agents recommend vasopressors, much more regularly as (SV)R (product of stroke volume and resistance) and mean blood pressure drop. This is consistent with physiological knowledge, and latest critical care research. For example, [40] shows that hemodynamic effects of norepinephrine extends beyond blood pressure, and it effects SV and CO, and as described earlier, increasing systemic vascular resistance and blood pressure, are among the primary goals of vasopressor therapy.

However, it is interesting to note that the clinicians have not necessarily associated lower blood pressure, or (SV)R with more frequent vasopressor administration. However they do seem to give vasopressors more regularly as SOFA score increase. These results could potentially provide an important direction and hints towards *better* treatment strategies.

This difference between the AI agent and human physicians is not unexpected, and does not imply that physicians are systematically acting sub-optimally. Rather, this difference reflects the fact that the rewards that the agents were trained on only consider the final state of the patient. They do not, for example, incorporate decisions that were made by the patient's family to cease extraordinary measures, after consultation with the physician. Such status changes are common, but were not available in the training data.

In contrast to vasopressors, RL agents and clinicians had similar frequencies of fluid administration for non-survivors. However, there were some disagreement even amongst the ensembles on whether or not to administer fluids for survivors (at less risky states). We present a more detailed analysis along with global results in the supplementary information (Appendix C in S1 text).

5.2.4 Uncertainty Aware Treatment

Next, we consider representative patients, and analyze the expected values of all distributions and model uncertainty. Fig 6A shows the evolution of expected values for a non-survivor (ICU ID: 263969). This was typical among all non-survivors; initially there's less variability among the expected values, but as the patient's health deteriorates the variation becomes more drastic, and there is a clear preference towards vasopressor-based actions. The marker size indicates how much the agent is uncertain of its own results. We observe that the model is less certain when the patient's health starts deteriorating. This can be attributed to the fact that these states are uncommon in the training data, and that the underlying cause driving deterioration can vary widely in septic patients.

For comparison, Fig 6B shows the expected values of a survivor (ICU ID: 279413). Here the expected values take a downward slide at around 25 hours from admission, with the values associated with no treatment considerably lower. This coincides with SOFA score

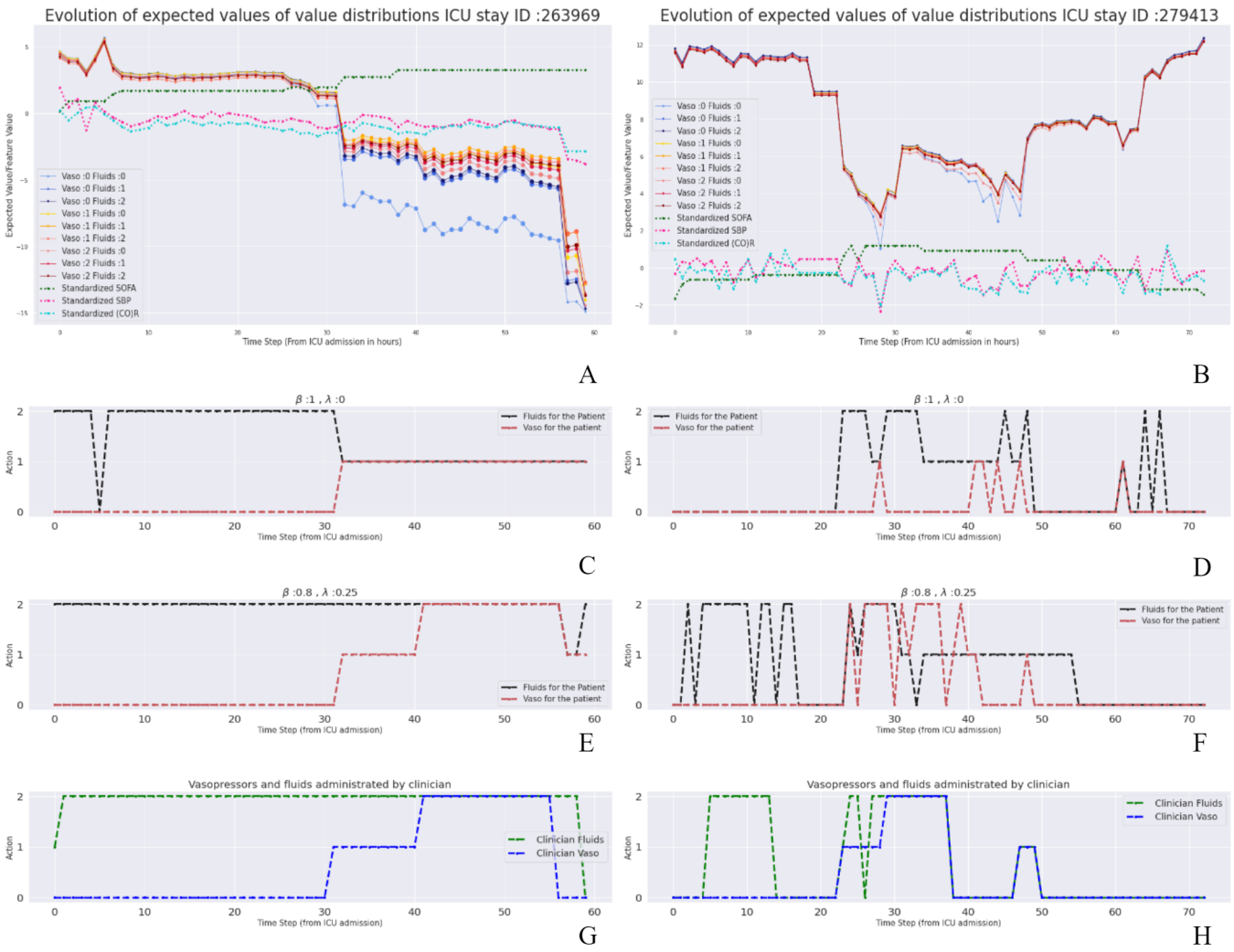


Fig. 6: **Expected value evolution of the main agent for two patients: (A)** A patient who died in the ICU. **(B)** A survivor. The marker size indicates the parametric uncertainty associated with a particular action. Also shown are the standardized values of SOFA score, Systolic blood pressure, and the unidentifiable cardiovascular state (CO)R. The x-axis indicates the hours from ICU admission. **Recommended treatments under various preference parameters:** (see Eq. 28). **(C)(E)** Recommendations for the same patient as in (A). **(D)(F)** Recommendations for the same patient as in (B). **Actual clinician treatments: (G)** treatment for the patient in (A), **(H)** treatment for the patient in (B).

increasing and SBP (CO)R rapidly decreasing, clearly indicating that the patient's health is deteriorating. However, as SOFA score improves and the pressure and (CO)R goes up, the

expected values do go up, and the difference between expected values of each distribution is considerably less. The uncertainty levels are also much lower.

The fact that expected values of different actions are close to each other in *healthy* patient states can be explained by equation 23. State-action values are calculated under the assumption that the agent always takes the optimal action. Our agent chooses an action every hour, and the intermediate rewards are much smaller in value than the terminal rewards. Thus, the value of the choice of action is not likely to change very much in a *healthy* patient state from hour to hour. Put another way, any mistake made by the agent is easily reversed by taking the correct action in the next hour if the patient is non-critical. In contrast, in more critical states, a wrong action can have irreversible consequences.

Fig 6C-Fig 6F show different treatment recommendations under our proposed framework for uncertainty-aware decision support. Briefly, the user specifies their relative confidence in the RL-agent and a behavior cloner (which represents the human agent) by specifying a parameter, β. Lower values of β place more emphasis on the behavior cloner. An action preference score (see. Methods, Eq. 28) is then calculated for each action in the current state. The score is a simple mixture of the scaled (using a softmax function) expected value of the ensembled distribution and the behavior probability, discounted by the model uncertainty corresponding to the state-action pair, using a parameter λ. Panels C-F illustrate that different choices are made, depending on the value of β and λ. Further, the sequence of treatments are qualitatively different for the non-survivor (panels C and E) and the survivor (panels D and F), because the agent has learned to identify critical states that require interventions; the average non-survivor tends to remain in such states for longer stretches, and so the agent makes relatively few adjustments, compared to the survivor. Once again, the agent does not know the ultimate fate of the patient. For comparison, panels G and H show the actual clinician treatments for the two patients.

5.2.5 Uncertainty Quantification Results

We now, briefly mention some interesting results on both model and environment uncertainties. Further results are available in the supplementary information (Appendix C in S1

text).

Fig 7A and Fig 7B present how model uncertainty changes with time to death and release for non-survivors and survivors respectively. It is interesting to note that on average the model is a lot more uncertain about non survivors compared with survivors. Further, as a patient gets *closer* to death the uncertainty increases, whilst for survivors the model uncertainty decreases closer they are to ICU release. This observation is not surprising since death states are relatively uncommon, and also there are a wide variety of ways a septic patient may face increased mortality risk. However for survivors, we do expect all of them to approach a *healthy* state as they approach eventual discharge.

Fig 7C and Fig 7D show the average entropy of the value distributions for each of the actions (again with time to death and release). This can be interpreted as a form of inherent environment uncertainty over future rewards. Now, there is less of a difference between the survivors and non-survivors and we can see a drastic drop in entropy for non-survivors as they approach death. This is not unexpected as the environment uncertainty should reduce when a patient's state has deteriorated beyond a certain point. Similarly the entropy of value distributions reduce for survivors nearer they are to release. It is also interesting to note that on average vasopressor based actions have a lower model uncertainty but a higher entropy.

Next we fit a Gaussian Mixture model for the data, and examined the model uncertainty with the predicted likelihood. Fig 7E shows the how the average (across all actions) model uncertainty for each data-point with a likelihood less than p th percentile. As one could expect the model uncertainty is higher for data-points with low density and reduces as the likelihood increases. This shows how the networks are uncertain of data away from the training distribution, and the value of having a large representative dataset.

5.2.6 A Comment on Off Policy Evaluation

Off policy evaluation (OPE) is the quantitative or statistical evaluation of the value of a learned policy, usually using another dataset. Although attractive in theory, most unbiased OPE methods use importance sampling, and are therefore dependent on a known behavior policy. This is not the case when the data were generated by human clinicians. Even if

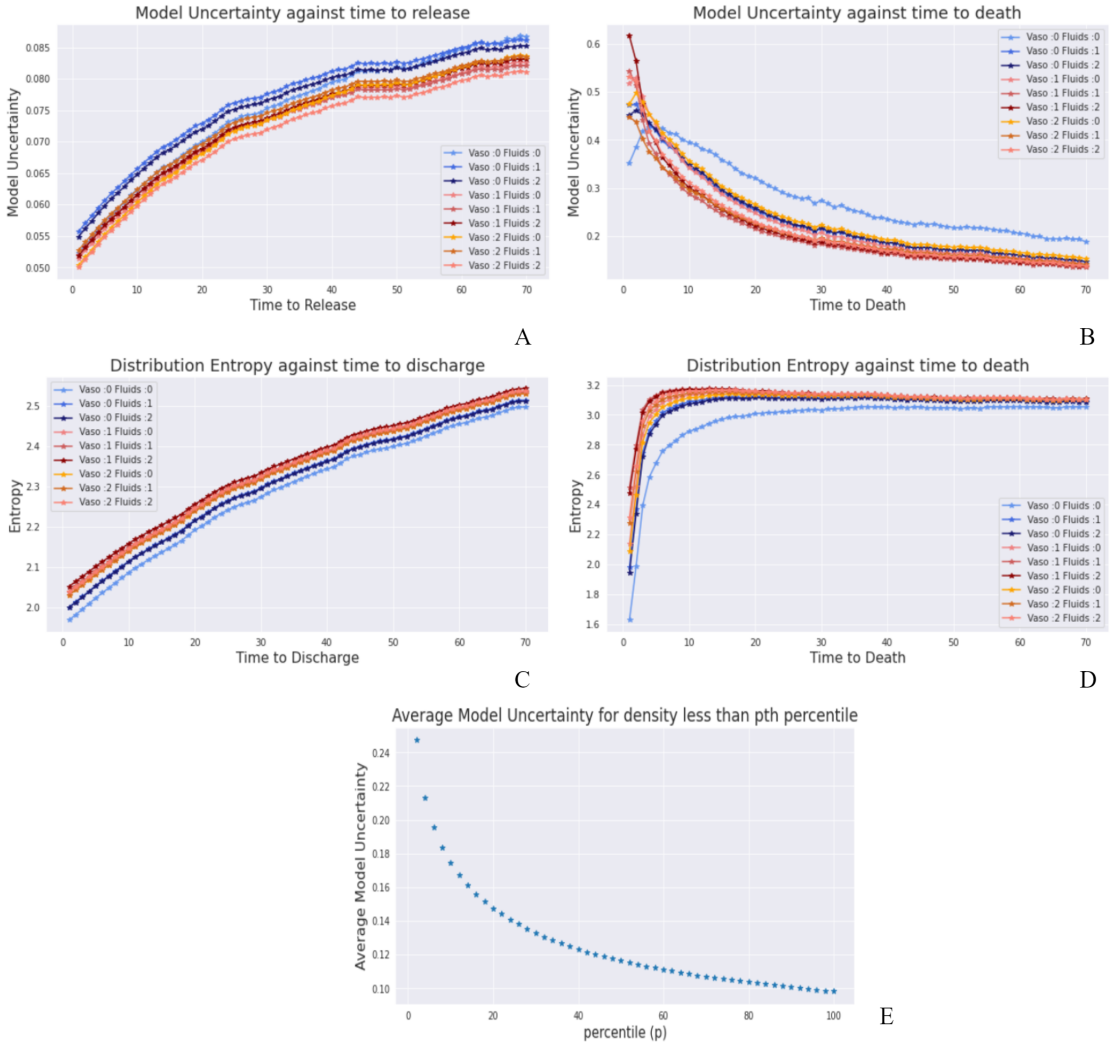


Fig. 7: **(A)** Model Uncertainty with time to death for non-survivors, **(B)** Model Uncertainty with time to discharge for survivors **(C)** Averaged entropy of value distributions for non-survivors with time to death, **(D)** Averaged entropy of value distributions for survivors with time to release. **(E)** Average Model Uncertainty for data points with density less than the p-th percentile.

a suitable behavior policy *were* known, an obviously bad policy can result in a very high OPE value in our setting. For example, an agent that always prescribes no treatment for critical patients would, in effect, eliminate most of the rewards accumulated by non-survivors which are, of course, the source of the majority of the *negative* rewards. Such a policy would have a misleadingly high OPE, because human clinicians rarely withhold treatment for critical patients (the one exception being a conscious decision by the family to terminate

extraordinary interventions), and so the the importance weights for such trajectories will tend towards 0.

We note that previous research has also argued at *all* OPE methods are unreliable in the context of sepsis management, and state-of-the-art OPE methods may fail to differentiate between obviously good and obviously bad policies [46]. However, we mention OPE results in the supplementary material (Appendix C in S1 text). We do note that developing OPE techniques suited for the critical care domain is an important area of research to explore in the future.

5.3 Discussion & Conclusion

We present an interdisciplinary approach which we believe takes a significant step towards improving the current state of data-driven interventions in the context of clinical sepsis, in terms of improving both outcome and interpretability. Indeed, we believe that the maximum benefit of Artificial Intelligence applied to medicine is best realized through the integration of mechanistic models of physiology whenever possible, uncertainty quantification, and human expert knowledge into sequential decision making frameworks.

Our contribution improves the status quo in several ways. Compared to prior work, our approach deals with partial observability of data, yet known physiology, by leveraging a low-order two-compartment Windkessel-type cardiovascular model in the context of self-supervised representation learning. As mentioned previously, this has several benefits. First, in the context of sepsis treatment, estimating the cardiovascular state is essential because the clinical decision to administer intravenous fluids or vasopressor is driven by an implicit differential diagnosis by the clinician, as to whether insufficient organ perfusion and shock are secondary to insufficient circulating volume (thus requiring fluids), vasoplegia (thus requiring vasopressors), or some combination of both fundamental pathophysiologies. Second, there is typically insufficient data to determine whether heart function is adequate (contractile dysfunction), but a mechanistic model provides an indirect means for estimating cardiac function by imposing known physiology. Finally, the incorporation of physiologic models

improves model explainability, while deep neural networks and stochastic gradient-based optimizers make it possible to learn robust and generalizable representations from large data. We expect the unification of models based on first-principles and data-driven approaches will provide a powerful interface between traditional computational sciences and modern machine learning research, mutually benefiting both disciplines. We have not fully examined the association between inferred physiological state and treatment recommendation to confirm whether recommended actions are indeed clinically sensible. Such work is currently underway.

We also introduce an approach to quantifying model uncertainty, which is essential in any practical application of RL-based inference using clinical data. To the best of our knowledge, this is the first time uncertainty quantification is used to quantify epistemic uncertainty in RL-based optimization of sepsis treatment, and of critical care applications more generally. (Previous approaches ex. [74] have considered inherent environment uncertainty). The method's uncertainty estimates, combined with the recommended action comprise a simple framework for automated clinical decision support. This principle aligns with the larger goal of combining different forms of expertise and knowledge for better decision making, a philosophy consistent with the rest of this work.

We chose a decision time step of one hour. Compared to similar work, this is much more compatible with the time scale of medical decision making in sepsis, where fluid and vasopressor treatments are titrated continuously. Accordingly, on such a time scale, there does not appear to be large differences in the relative merit of different dosing strategies. This makes intuitive sense: there is presumably a lesser need for major treatment modifications if decisions are made more frequently. Yet, a frequent finding across patients, especially the sickest ones, was that inaction (no intervention) was a consistently worse strategy. This also meets clinical intuition.

Reducing the time scale of decisions is not only appealing clinically in situation of rapidly evolving physiological states, such as is the case in early sepsis, but it also provides a more compelling basis for a less granular action space. Indeed, if decisions are made hourly, it does meet clinical intuition to have fever discrete actions. Few physicians will argue that there is likely to be little difference in administering 100cc or 200cc of fluids in the next hour. In the extreme, if time were continuous, the likely decision space at any given time, is whether

a fluid bolus should be administered or not. A similar reasoning applies to vasopressors (increase, reduce, status quo). We further notice that our methods consistently identify high risk, non-survivor patient states which can *potentially* benefit from more frequent vasopressor treatment. These results should of course, be subject to clinical verification.

An important open problem in the application of offline RL to medicine is the means by which one evaluates learned treatment policies, given the obvious ethical issues associated with allowing an AI to exert some control over treatment. Still, proper clinical trials will be necessary, eventually, so the critical care community should define for itself the standards by which an AI would be deemed safe enough to enter clinical trials [105]. In this work, we have largely relied on a combination of medical expertise, and the fact that our model leverages prior knowledge in the form of a simple model of cardiovascular physiology, to argue that the learned policy is reasonable. We make no claim that the policy is expected to produce superior outcomes in sepsis patients, relative to human clinicians. One important area for future work may be the incorporation of more detailed models of physiology into our framework, or perhaps using such models in the context of *in silico* trials (ex. [31]) as a first step towards demonstrating that a learned policy is safe, and perhaps suitable for pre-clinical and clinical trials. Additional areas for future work include the design of alternative rewards (ex. based on time-dependent hazard ratios for death), and the application of risk-averse offline RL (ex. [121]).

5.4 Methods

5.4.1 Data sources & Preprocessing

Our cohort consisted of adult patients (≥ 17) who satisfied the Sepsis 3 [55] criteria from the Multi-parameter Intelligent Monitoring in Intensive Care (MIMIC-III v1.4) database [56], [93]. We excluded patients with more than 25% missing values after creating hourly trajectories, and patients with no weight measurements recorded. The starting point of trajectories is ICU admission.

We further excluded patients who got discharged from the ICU but ended up dying a few days or weeks later at the hospital. Since we don't have access to their patient data after the ICU release, treating the final ICU data as a terminal state would damage generalizability. We cannot treat those patients as survivors, however, as they were not released from the hospital.

Actions were selected by considering hourly total volume of fluids (adjusted for tonicity), and norepinephrine equivalent hourly dose (mcg/kg) for vasopressors. In computing the equivalent rates of each treatment, we followed the exact same queries as Komorowski et al [63]. When different fluids were administrated, we summed up the total fluid intake within the hour, and discretized the resulting distribution. For vasopressors, we considered the maximum norepinephrine equivalent rate administered within the hour to infer the hourly dose. We used 0.15 mcg/kg/min norepinephrine equivalent rate, and 500 ml for fluids, as the 1,2 cutoff when discretizing. These were chosen, considering the mean, median of non zero rates and medical knowledge, We also observe that due to the low dimensional action space, there is flexibility in choosing the cutoffs. A separate 0 action for each was added to denote no treatment.

Missing vitals and lab values were imputed using a last value carried forward scheme, as long as missingness remained less than 25% of values. A detailed description on extracting, cleaning and implementation specific processing as well as additional cohort details are included in the supplementary information (Appendix A and Appendix B in S1 text).

5.4.2 Models

5.4.2.1 Physiology-driven Autoencoder

Autoencoders are a type of neural networks which learn a useful latent, typically lower-dimensional representation of input data, while assessing the fidelity of this representation by minimizing data reconstruction error. Our autoencoder architecture provides an implicit regularization by constraining the latent states to have physiological meaning, and the decoder to be a fixed physiologic model described in the next section. We further use a denoising scheme by randomly zeroing out input with a probability of 10-25% , when feeding into the

network. This random corruption forces the network to take the whole patient trajectory (prior to the current time point) and previous treatment into account when producing its output, because it prevents the network from *memorizing* the current observation. In essence, we ask the inference network to predict observable blood pressures and the heart rate using corrupted versions of itself, by first projecting it into the cardiovascular latent state, and then decoding that to reconstruct.

More precisely, at time t, suppose the full history up to and including t is represented by h_t. Then, the output of the system $\hat{o_t}$ satisfies,

$$\hat{o_t} = f(g(\tilde{h_t}, a_t, d)).$$

Here $\tilde{h_t}$ is the corrupted history computed as,

$$\tilde{h_t} = h_t(\odot)p$$

where p is a vector of same dimensions as h_t such that each element is sampled independently from a Bernoulli distribution, and $(\odot)$ denotes element wise multiplication. g, f are the encoder and the decoder respectively, a_t denotes the treatment at time t and d denotes the demographic variables. The decoder f is detailed out in the next section, and g is the composition of neural networks as shown in Fig 1.

Fig 1C, shows the complete architecture of our inference network. As shown in the figure, the encoder is comprised of three neural networks, a patient encoder which computes initial hidden state estimates, a gated recurrent unit (GRU) [29] based recurrent neural network to encode the past history of vitals and scores up to and including the current time point, and a transition network which takes the previous state, the action and the history representation to output new cardiovascular state estimates. We train this structure end-to-end by minimizing the reconstruction loss, using stochastic gradient-based optimization. The supplementary material (Appendix B in S1 text) provides a detailed description of model and architecture hyper-parameters, and training details.

The cardiovascular model, is based on a two-element Windkessel model illustrated using the electrical analog in Fig 1B. This model provides a lumped representation of the resistive and elastic properties of the entire arterial circulation using just two elements, a resistance

R and a capacitance C, which represent the systemic vascular resistance (SVR), and the elastance properties of the entire systemic circulation, respectively. Despite it's simplicity, this model has been previously used to predict hemodynamic responses to vasopressors [16] and as an estimator of cardiac output and SVR [17].

The differential equation representing this model is:

$$\frac{\mathrm{d}P(t)}{\mathrm{d}t} = -\frac{1}{RC}P(t) + \frac{Q(t)}{C} \tag{24}$$

where $Q(t)$ represents the volume of blood in the arterial system. As explained in [16], over the interval $[0, T]$ (where T is the filling time of the arterial system) we can write $Q(t)$ as $Q(t) = SV\delta(t)$, where SV stands for Stroke Volume, the volume of blood ejected from the heart in a heartbeat. When the system is integrated over the interval $[0, T]$ we obtain the following expressions for $P_{sys}, P_{dias}, P_{MAP}$, i.e., the systolic,diastolic, mean arterial pressure, respectively,

$$P_{sys} = \frac{SV}{C}\frac{1}{1 - e^{-T/RC}}, \quad P_{dias} = \frac{SV}{C}\frac{e^{-T/RC}}{1 - e^{-T/RC}}, \quad P_{MAP} = \frac{(SV)R}{T} = \frac{(SV)FR}{60} \tag{25}$$

T is the filling time and F is the heart rate, which is determined by T. This system of algebraic equations is used for the decoder of our autoencoder. Since heart rate can itself be affected by vasopressors and fluids, we added heart rate (F) as an additional cardiovascular state despite it being observable.

Therefore we have a multivariate function $f : \{R, C, SV, F, T\} \to \{P_{sys}, P_{dias}, P_{MAP}, F\}$, represented by the equations above, and the trivial relationship $F = F$ (Despite the obvious relationship we used both F and T, for ease of training and stability.) As stated previously, to prevent it from just using the current observations, we use a denoising scheme for training. This ensures at a fixed time, the model cannot *memorize* the current observation and learn to invert f, since there is a nonzero probability of corruption. Thus it has to learn to factor in the history and the treatments when determining the cardiovascular states. Once SV is inferred, the cardiac output (CO), can be computed as $CO = (SV)F$.

Since f is not one to one, typically not all states are identifiable. To arrive at a better approximation we used the latent space to only model deviations from fixed baselines. We

also posit that identifiable combinations of states, when trained with a denoising scheme, should provide important cardiovascular representations in the POMDP setting.

5.4.2.2 Denoising GRU Autoencoder for Representing Lab History

We use another recurrent autoencoder to represent patient lab history, motivated by the fact that labs are recorded only once every 12 hours. Forward filling the same observation for 12 time points, is almost certainly sub-optimal, and the patterns of change in lab history can be helpful in learning a more faithful representation. Thus, we use a denoising GRU autoencoder constructed by stacking three multi-layer GRU networks on top of each other, with a decreasing number of nodes in each layer, the last 10 dimensional hidden layer was used as our representation. This architecture is motivated by architectures used in speech recognition [24].

This model was also trained by corrupting the input, where each data-point was zeroed with a probability of up to 50%. (The rate was gradually increased from 0 to 50%). As with the previous autoencoder, this provides an extra form of regularization, and forces the learned representation to encode the entire history.

Model architecture and training details and presented in the supplementary materials (Appendix B in S1 text).

5.4.2.3 Behavior Cloner

We use a standard multi-layer neural network as our imitation learner. This model is trained using stochastic gradient-based optimization by minimizing the negative log-likelihood loss, between the predicted action and the observed clinician action, with added regularization to prevent overfitting.

We do mention that there are many other options that could be used as a imitation learner, including nearest neighbor-based method as in [92].

5.4.3 POMDP Formulation

A state is represented by 41 dimensional real-valued vector consisting of:

- **Demographics**: Age, Gender, Weight.
- **Vitals**: Heart Rate, Systolic Blood Pressure, Diastolic Blood Pressure, Mean Arterial Blood Pressure, Temperature, SpO2, Respiratory Rate.
- **Scores:** 24 hour based scores of, SOFA, Liver, Renal, CNS, Cardiovascular
- **Labs:** Anion Gap, Bicarbonate, Creatinine, Chloride, Glucose, Hematocrit, Hemoglobin, Platelet, Potassium, Sodium, BUN, WBC.
- **Latent States:** Cardiovascular states and 10 dimensional lab history representation.

To ensure each action has a considerable representation in the dataset, we discretize vasopressor and fluid administrations into 3 bins, instead of 5 as in previous work [99], [63] [92]. This results in 9 dimensional action space.

1 hour

We use the reward structure that was suggested by Raghu et. al [99], with a minor modification. Since lactate was very sparse amongst out cohort we only considered SOFA based intermediate rewards. Specifically, whenever s_{t+1} is not terminal, we use reward of the form:

$$r(s_t, a, s_{t+1}) = -0.025\mathbb{I}((s_{t+1}^{SOFA} = s_t^{SOFA} \,\&\, s_{t+1}^{SOFA} > 0) - 0.125\mathbb{I}(s_{t+1}^{SOFA} - s_t^{SOFA}) \qquad (26)$$

For terminal rewards we put $r(s_t, a, s_{t+1}) = 15$ for survival and $r(s_t, a, s_{t+1}) = -15$ for non-survival.

5.4.3.1 Training

We only mention important details of training the RL algorithms here. Representation Learning related training and implementations are detailed out in the supplementary information (Appendix B in S1 text).

We train the Q networks using a weighted random sampling-based experience replay, analogous to the prioritized experienced replay [108], which has resulted in superior performance in classical DRL domains, such as Atari games.

In particular for each batch, we sample our transitions from a multinomial distribution, with higher weights given to terminal death states, near death states (measured by time of eventual death), and terminal surviving states. We used a batch size of 100, and adjusted weights such that on average there is 1 surviving state, and 1 death state in each batch.

This does introduce bias, with respect to the existing transition dataset, however we argue that this would correspond to sampling transitions from a different data distribution, which is closer to the true patient transition distribution, we are interested in, as we are necessarily interested in reducing mortality. We empirically observe that, when using such a weighting scheme the value distributions align more closely to clinical knowledge in identifying risky states, and *near* death states.

A same weighting scheme was used for all ensemble networks, which are trained to estimate uncertainty. As mentioned previously, we verify that the main results on vasopressor treatment strategies hold even for pure random sampling.

5.4.4 Uncertainty

In this section, we consider model uncertainty, and not the inherent environment uncertainty. Model uncertainty stems from the data used in training, neural network architectures, training algorithms, and the training process itself.

Inspired by statistical learning theory [123], and the associated structured risk minimization problem [134], we define the model uncertainty, (conditioned on a state s and a action a), given our learning algorithm, and model architecture as :

$$\mathbb{E}_{\theta,D}[l(\theta, \mathbb{E}_D[\theta])|s,a] = \int l(\theta, \mathbb{E}_D[\theta])|_{s,a} p(\theta, D) d\theta dD = \int l(\theta, \mathbb{E}_D[\theta])|_{s,a} p(\theta|D) p(D) d\theta dD \tag{27}$$

Here, D denotes the unknown distribution of ICU patient transitions that we are attempting to learn our policies with respect to. θ is a random variable which characterizes

the value distributions. (For the C51 algorithm this can be interpreted as an element in $\mathbb{R}^{51}$). This is outputted by our networks trained on a dataset sampled from D, for a given state action pair. This random variable is certainly dependent on the training data, and the randomness stems from the inherent randomness of stochastic gradient based optimization [62] and random weights initialization. The quantity l is a divergence metric appropriate for comparing probability distributions. We use the Kullback–Leibler divergence [66] for l.

5.4.4.1 Estimating the Uncertainty Measure

We construct a Monte-Carlo estimate of the integral in (27) by bootstrapping 25 different datasets, each substantially smaller than the full training dataset, and training identical distributional RL algorithms in each. This can be done efficiently due to the sample efficiency of distributional methods. Additionally, we can approximate $\mathbb{E}[\theta]$ either by the ensemble value distribution, or by the value distribution of the model trained on the full training dataset.

5.4.5 Uncertainty Aware Treatment

In this section, we describe a general framework for choosing actions that factors in uncertainty. Notice that, because our RL algorithm learns (an approximation of) the optimal value distributions, making decisions by considering additional information does not violate any assumption underlying the learning process.

When suggesting safe treatment strategies, we want the proposed action to have high expected value, however we would also like our agent to flexible enough to propose an action with less model uncertainty, if two actions have very close expected values to each other. Another important factor to consider is how likely an action is to be taken by a human clinician. This will have significance in a situation where human expertise is scarce. Large retrospective datasets subsume experience of hundreds of clinicians, and knowing what previous clinicians have done in similar situations, will be valuable such situations. Therefore we use behavior cloning to learn an approximate behavior policy of clinicians on average.

To satisfy all three goals, we propose a general framework for choosing actions, based on an action preference score, $\mathcal{P}(s, a)$, parameterized by two parameters. This general framework

is flexible, yet simple, and the end-user can choose the parameters to reflect their own expert knowledge, and confidence of the framework.

Let $G(s, a)$ be a human behavior likelihood score function. In this work we equate $G(s, a)$ with the probabilities outputted by the behavior cloning network described in section 6.2.3. Given a state s, we define $\mathcal{P}(s, a)$ associated with each action a, as:

$$\mathcal{P}(s, a) = \beta(Softmax(\tilde{Q}^*(s, a)) + (1 - \beta)G(s, a) - \lambda u(s, a) \tag{28}$$

where $\beta, \lambda \geq 0$, $u(s, a)$ is the parametric uncertainty associated with the state-action pair, s, a, $G(s, a)$ is the behavior likelihood probability and $\tilde{Q}^*(s, a)$ is the Q function computed from the ensembled value distributions. When human expertise is available, $G(s, a)$ can be modified or even re-defined to factor in expert opinion. λ penalizes uncertainty, and a low β forces the action to be close to a clinician action. We could recover the expected value criteria by setting $\beta = 1, \lambda = 0$, and we could use the system as a pure behavior cloner, by setting $\beta = 0, \lambda = 0$. Therefore β controls how far from the highest expected value/behavior likelihood score can the agent choose an action.

5.5 Supplementary Information

5.5.1 Appendix A: Cohort Details

Our total patient cohort consists of 18,472 patients, out of which 1,828 were non-survivors.

Table 2: Cohort details

Cohort	% Female	Mean Age	Mean ICU Stay	Total Population
Overall	42.33 %	66.05	7 days 15 hours	18472
Non-Survivors	42.67 %	68.8	9 days 13 hours	1828
Survivors	42.14 %	65.91	5 days 13 hours	16644

This resulted in an experience replay consisting a total of 2596604 transitions.

5.5.2 Appendix B: Neural Network Architectures and Implementation Details

5.5.2.1 Physiology-driven Autoendcoder

Encoder :

- Patient Encoder: Multi-layer feed-forward neural network, with 3 hidden layers with 64 nodes each, followed by exponential linear unit, (eLU) non-linearity applied element wise.
- Transition: Multi-layer feed-forward neural network, with 8 hidden layers with 128 nodes each, followed by exponential linear unit, (eLU) non-linearity applied element wise.
- RNN: Gated recurrent unit, based RNN, with 1 hidden layer, with 64 nodes.

We note that, as inputs for the network specifically the RNN, we included all vitals, and SOFA-related scores, including the four dimensional observations, systolic blood pressure, diastolic blood pressure, mean blood pressure and heart rate.

For training, we used Adam [62], with a low learning rate (1e-5), the corruption was only introduced after the model has been trained for several epochs. For RL representation we used the model trained with 10% corruption.

5.5.2.2 Denoising Lab Autoencoder

This is comprised of three GRU networks stacked on top of each other.

- Network 1 : 12 hidden units, with 512 nodes, outputs a 128 node vector.
- Network 2 : 5 hidden units 128 nodes each, outputs a 10 dimensional vector.
- Network 3: 3 hidden units, with 10 nodes each, the last of which is taken as our hidden lab representation.

We train this again using Adam, and corruption is gradually introduced starting from 0% to 50%. We use the network trained under 50% corrupted inputs, when inferring the hidden lab representation for RL.

We standardized all the labs before feeding into the network.

5.5.2.3 Imitation Learning

Muli-layer neural network with 4 hidden layers: 3 with node size 512, and the last 256. All hidden (and input) layers are followed by a rectified linear unit (reLU) non-linearity.

Training was again using Adam with a standard learning rate, and we minimized a negative log-likelihood loss, which is standard in classification problems.

5.5.2.4 Bootstrapping and Deep Ensembles

To learn each bootstrapped network, we first sampled from the all patients to arrive at a bootstrapped patient list. Then we train the networks, for 2 or 3 epochs each (to have further randomness), using a process identical to training the main RL algorithm.

For the uncertainty quantification step, we trained the majority of bootstrapped ensembles on as little as 40% of total patients, however for the results presented under vasopressor administration, we only considered ensembles which were trained on a cohort of 65-80% of patients. The number of patients was also picked at random. There were 20 such bootstrapped ensembles.

5.5.2.5 Distributional Q learning

We use the standard C51 training algorithm as in [10]. Q network was a multi-layer neural network. Apart from the weighted sampling described in the main body of the text, training steps were all standard.

We use a target network, and update the target networks using polyak target updating with $\tau = 0.005$. (i.e. after every iteration/training step we set the target network weights to a linear combination of it's own weights, weighted by (1-τ) an the Q network weights, weighted by τ). This kind of target network is common amongst all deep Q learning, algorithms.

We summarize the hyper-parameters involved in table below.

Table 3: RL algorithm hyper-parameters

Hyper-Parameter	Value
Support size	51
Maximum value	18
Minimum value	-18
γ	0.999
Batch size	100
Number of iterations	51932
Optimizer	Adam
Learning rate	$3 * 10^{-4}$
τ	0.005

5.5.3 Appendix C: Additional Results

5.5.3.1 RL Results

In this section, we present further results of the distributional RL algorithm, and un-certainty quantification. First, in Fig 8 we present the feature importance of all features.

Fig 9 shows, expected value trajectories, of randomly selected validation patients. As we have mentioned previously these results indicate the generalizability of our value networks.

To be consistent with previous work (e.x. [99]), we present heat plots of global actions, overall, low SOFA (< 5), medium SOFA ($\geq 5 < 15$) and high SOFA (> 15) separately. We

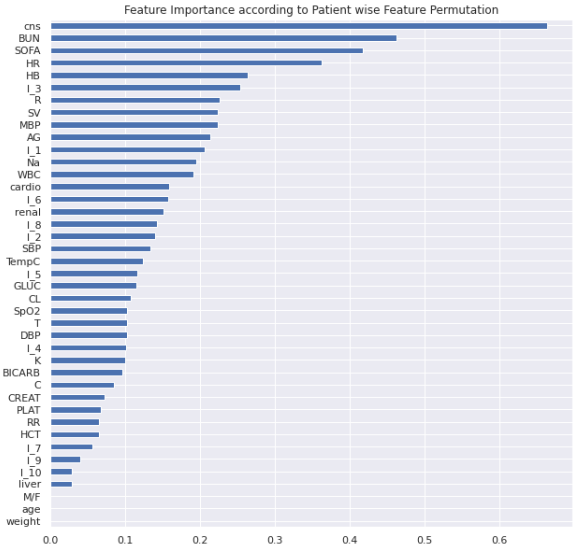


Fig. 8: Feature Importance measured by feature permutation. Here, l_k denotes the kth component of the latent lab representation

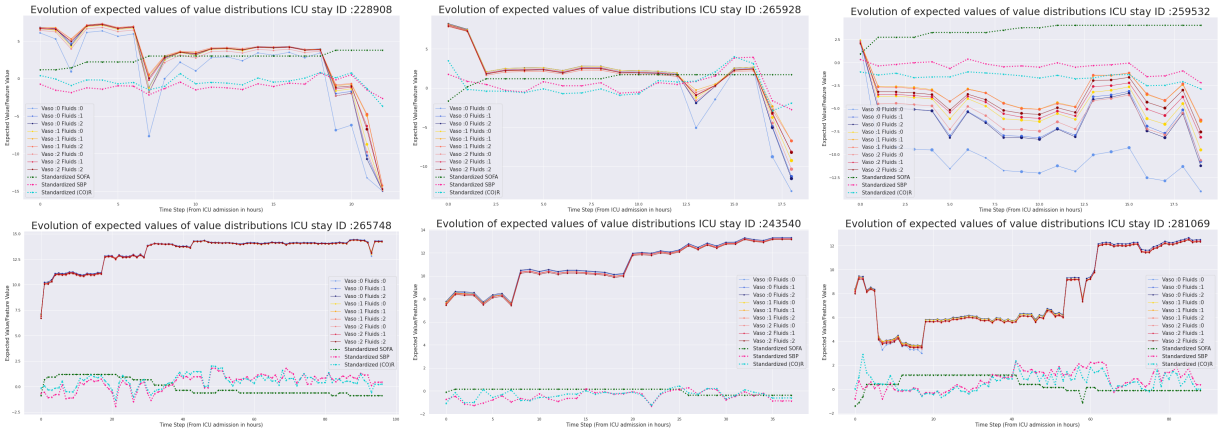


Fig. 9: Expected Values of random validation patients, **Top**: Non-survivors, **Bottom**: Survivors. As with Fig 4, the blob size indicate the uncertainty

further provide last 24 hours on non-survivors and results from decisions taken with respect to the expected value of the an ensembled weighted distribution, (corresponding to $\beta = 1, \lambda = 0$) and $\beta = 0.8, \lambda = 0.25$.

However, at its core, our approach strives to extract and use patient specific recurrent

representations to learn personalized treatments. Therefore global analysis is unlikely to provide much insight into the intricacies that underline the decision making process. Further when analyzing the proposed treatment, it should be noted that each action is proposed considering only the current, actual state. Therefore for a fixed patient trajectory at a fixed time, the agent does not know what it has proposed previously, nor how its action would have impacted the state.

The most striking difference is for non survivors near death states. Our methods consistently recommend vasopressors. It is also interesting that RL methods have in general preferred low/medium (corresponding to 1) vasopressors and fluids as opposed to high doses (2). Just as we mentioned in the main text, when ensembled, agents do not recommend fluids for survivors' less critical states. It must be noted however that the agent trained on the whole cohort did have fluids recommended regularly, but there is disagreement amongst the ensembles.

5.5.3.2 Uncertainty Quantification Results

In this section, we briefly mention results of uncertainty quantification.

The common pattern is that for most patients who have died, the model is less confident about its value distributions as they become closer to death. Uncertainty among each action varies from patient to patient. However for survivors this behavior is the exact opposite, as the agent is more confident of its results and becomes even more confident as the patient gets closer to discharge from the ICU. We illustrated this in Fig 7 in the main text, which presented averaged model uncertainty with time to death and discharge for non-survivors and survivors respectively.

Table 4 presents average uncertainty, among all patient states, grouped by the training and validation datasets and whether the patient was a survivor or a non-survivor. As we mentioned in the main text, the uncertainty is much higher for non survivors than survivors. Further uncertainties for validation non-survivors are higher than training non-survivors. However for survivors the training and validation uncertainty are very similar on average.

Both Table 4 and Fig 7 agree with our expectations, because near-death states, are

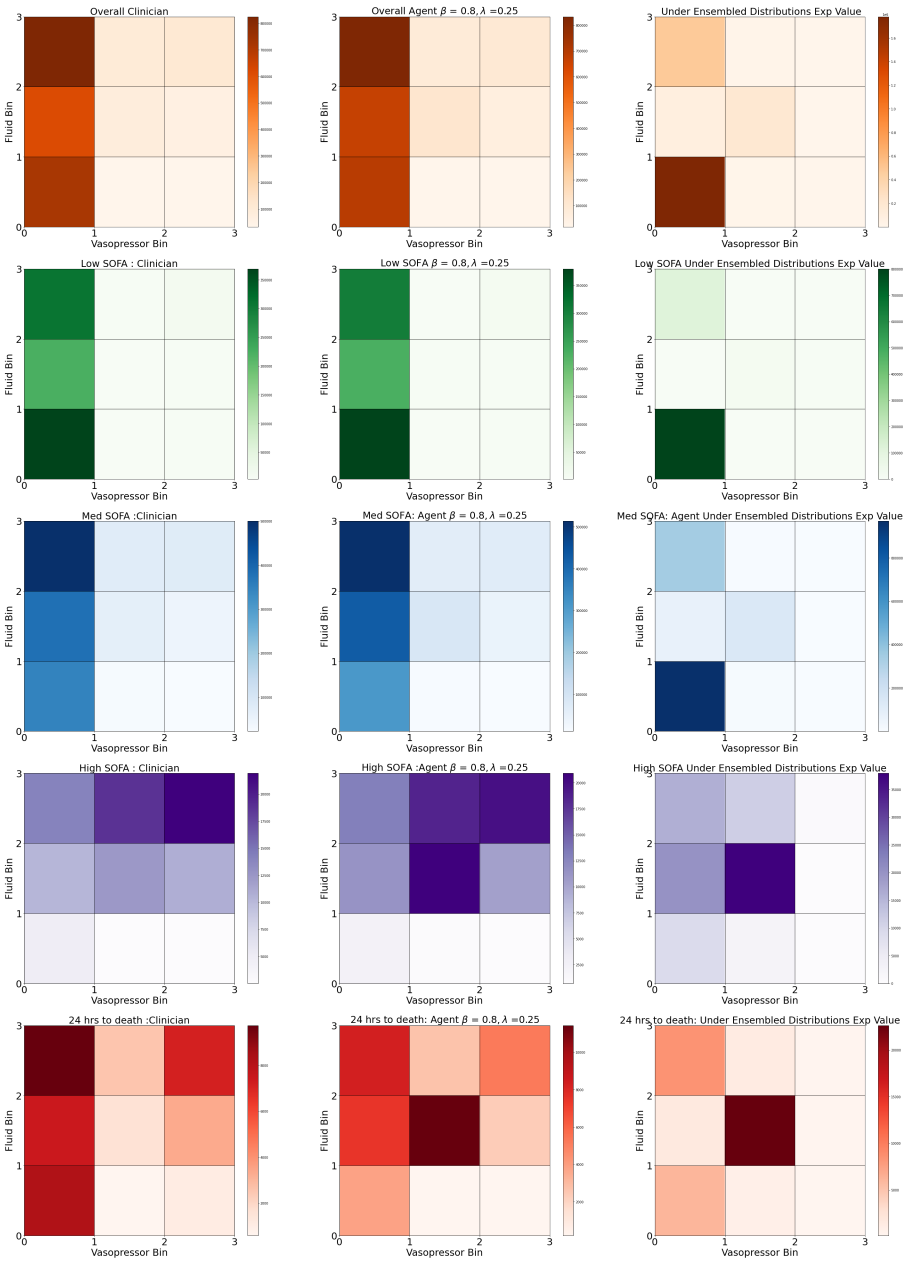


Fig. 10: **L**: Heat-plots for recommended actions, under $\beta = 0.8$, $l\lambda = 0.25$ and Ensembled Distribution Expected Values. Shown are clinician's vs Agent for overall (orange), low sofa (green), medium sofa (blue), high score (purple) and non survivors last 24 hrs (red).

relatively uncommon, and also there could be a lot of different ways a septic patient may have increased mortality risk. However, for survivors, we do expect our agent to be confident of their survival, as their states should approach a *healthy* state.

Table 4: Mean model uncertainty for survivors and non-survivors in training & validation data

Action	Non-Survivors Train.	Survivors Train.	Non Survivors Val.	Survivors Val.
Vaso 0 Fluids 0	0.1916	0.0887	0.3617	0.0861
Vaso 0 Fluids 1	0.1591	0.0855	0.3085	0.0894
Vaso 0 Fluids 2	0.1587	0.0846	0.3104	0.0879
Vaso 1 Fluids 0	0.1547	0.0820	0.3066	0.0850
Vaso 1 Fluids 1	0.1451	0.0815	0.2676	0.0847
Vaso 1 Fluids 2	0.1482	0.0827	0.2776	0.0850
Vaso 1 Fluids 0	0.1634	0.0839	0.3135	0.0853
Vaso 2 Fluids 1	0.1498	0.0831	0.2710	0.0860
Vaso 2 Fluids 2	0.1488	0.0808	0.2850	0.0832

5.5.3.3 OPE Results

Despite the inherent limitations of OPE methods, we present results of Weighted Importance Sampling (WIS) OPE estimates of the validation cohort. Here, we compute the OPE estimates assuming our agent takes an action based on a score (either the expected value or the preference score in Equation 28) with a probability of 0.99 and takes a random action with a probability of 0.01. This was done to make the policy stochastic, because taking importance sample based estimates of deterministic policies can be problematic.

For a dataset D the WIS value estimate is computed as,

$$WIS(D) = \frac{1}{\sum_{i=1}^{n} w_i} \sum_{i=1}^{n} w_i (\sum_{t=1}^{L^i} \gamma^t (r_t^i)) \tag{29}$$

Where $w_i = \prod_{t=0}^{L^i} \pi_e(a_t|s_t)/\pi_b(a_t|s_t)$, a_i is the action taken in the dataset, π_e is the policy being valuated and π_b is the behavior policy. And L^i is the length of the trajectory of the i^{th} patient

As mentioned in the main text, Importance Sampling based OPE methods require a

known behavior policy. We estimate this by training a neural network as a behavior-cloner on the observable variables.

The clinicians' value estimate for the validation cohort was 12.44. The OPE value estimate for the ensemble agent taking actions with $\beta = 0.8, \lambda = 0.25$ was 13.03. The ensemble agent taking actions under expected value resulted in an OPE value estimate of 13.3. We further evaluated the value estimates on each of the bootstrapped ensemble. These numbers are shown as a box plot in Fig 11. Whilst all of the values were greater than the clinicians' value, for the reasons explained in the main text we note that these results don't necessarily imply that the RL agent is superior to the clinicians.

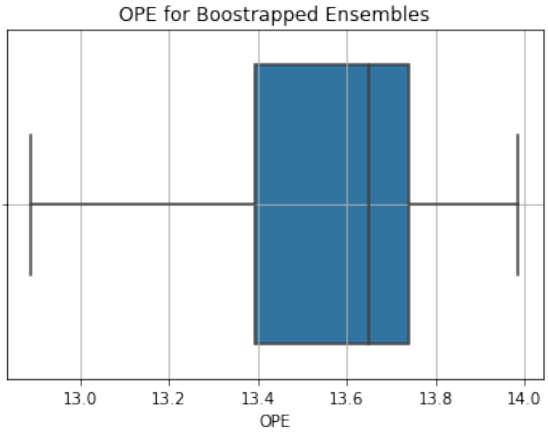


Fig. 11: Box plots of validation (weighted important sampling) OPE estimates for bootstrapped ensembles

5.5.4 Appendix D: Limitations and Open Problems

As stated in the main text, and discussed in previous work, the main limitation of *any* data driven or computational approach to finding optimal treatment is proper evaluation of the learned policy. In this work we relied on medical expertise and physiologic knowledge in interpreting the results, but evaluating learned policies is an active research area in offline RL, and future research could find better methods which are more suited to critical care applications.

A related issue is model selection. Like supervised learning, it has been shown previously

that training deep RL algorithms longer on the same dataset can result in poor performance and overfitting. A lack of an obvious evaluation metric (such as test accuracy for a classification problem) makes model selection complicated. We used results after only two full passes of the dataset (51932 iterations), observing that the results don't make the same sense, clinically, when it is trained for too long. Indeed Fig 12 shows the expected value evolution for a validation cohort non-survivor for different weights. As we can see, its results are far too optimistic when the patient is a few hours away from death, if trained longer. However, the vasopressor recommendation results for non-survivors, which was presented earlier, do hold for all the different training weights.

Fig. 12: Expected Values of non-survivor, **Left**: Trained for 2 epochs, **Right** : Trained for 7 epochs

In the context of sepsis treatment, as we mentioned under Discussion, a further challenge is designing rewards. For example even *survivors* have a high risk of relapse, and their physiologic age is significantly higher than their actual age. Therefore it can be argued that survival at the ICU should not be *rewarded* as much as (in absolute value) death. Further organ damage and mortality could be competing objectives for some patients. Whilst it is possible to have a weighted combination, of both as we did, a multi-objective RL framework could also be looked into. As we mentioned before we hope to explore these questions in future work.

5.5.4.1 Future Work

There are other avenues we would like to explore.

Model-based RL With Physiological Models: Model-based RL aims to explicitly model the underlying environment and then use this information in various ways for control.

[2] This paradigm provides a natural place to incorporate mechanistic models, which could potentially help both control and interpretability. Clearly, the availability of more granular data, or of additional domains of data, could allow better estimation of the underlying physiological model and thus reduce uncertainty.

Reward Structure: Our reward structure was based on previous work and has clinical appeal. However, rewards are an essential component of any RL algorithm and is the only place where the agent can judge the merit of its proposed actions. This is potentially another place to include physiological knowledge. Ideally, we would want our reward structure to capture an accurate mortality risk, and an organ damage score, with each state. Risk-based rewards, rooted in anticipated evolution over a meaningful clinical horizon, should be considered in future schemes.

Risk Averse RL: It could be argued that, maximizing the sum of expected future rewards may not best reflect the end goals of safety critical domains. Whist the rewards can be engineered to promote risk aversion, risk averse RL is a fast growing research area, which we are keen to explore, if the RL objective itself can be tweaked to be more suitable for critical care research.

[2]It could in theory be argued that our work itself is a model based and model free hybrid method.

6.0 Prologue to Article 2

The next article introduces a novel contrastive representation learning objective and a training scheme for clinical time series. Specifically, we project high dimensional EHR. data to a closed unit ball of low dimension, encoding geometric priors so that the origin represents an idealized *perfect* health state and the Euclidean norm is associated with the patient's mortality risk. Moreover, for septic patients, we show how we could learn to associate the angle between two vectors with the different organ system failures, thereby, learning a compact representation which is indicative of both mortality risk and specific organ failure. We show how the learned embedding can be used for online patient monitoring, supplement clinicians and improve performance of downstream machine learning tasks.

This work uses semi-supervised contrastive learning exploiting the underlying structure and regularity among critically ill patients.

Whilst this thesis is focused on sepsis and RL, the work presented in this article can be applied to any critically ill patient distribution, and has benefits beyond RL. However, the challenges of RL for sepsis motivated this work (The need to introduce a systematic way of defining intermediate rewards). Hence, we also show how such a design in terms of the learned embedding can result in qualitatively different policies and value distributions, compared with using only terminal rewards.

This work is joint work with supervisors Dr. Christopher James Langmead, Dr. Gilles Clermont, and Dr. David Swigon.

7.0 Article 2: Deep Normed Embeddings for Patient Representation

Recently contrastive methods, usually framed as self-supervised learning problems have enjoyed tremendous popularity and success across various domains [48, 26, 129], but their applications for electronic health record data have been limited [131]. Whilst this can be explained by complexity and noise in medical time series and the difficulty to create medically meaningful augmented versions of the patient states, there is an underlying regularity and structure amongst critically ill patients which we believe can be exploited, to produce a representation using simple geometric priors, working in the semi-supervised [1] setting instead of the fully self-supervised or supervised settings. For this purpose, we introduce a new optimization criteria, using which we embed high dimensional patient states to a lower dimensional unit ball. The embedding has the property that the mortality risk can be associated with the level sphere the embedded vector belongs to, and it can distinguish between variations and similarities between patients states subjected to the same mortality risk, using minimal supervision.

We evaluate our method on a large cohort of septic patients from the MIMIC-III [93, 56] database. Since our experiments are focused on septic patients, we encode similarly using major systems of organ failure. However, we note that the method can be easily adopted for any subset of patients who exhibit a few major, loosely defined physiological classes of criticality, and can approach higher mortality risks in different ways. By leveraging such basic medical knowledge, our method avoids the need to compute data augmentations to create similar pairs. Unlike in images, augmentations may not produce realistic patient states, due to the high complexity and correlations amongst the data dimensions, and the invariances amongst patient states are less clear. Therefore, we define similarities across two dimensions, a) mortality risk. b) major organ system failures (or a similar notion of similarity), and use a triplet based learning scheme, leveraging local stochastic gradient optimization. We illustrate our method using two simple network architectures a) an auto-regressive GRU network using

[1] Throughout, we use semi-supervised learning to mean learning with some form of partial supervision. We acknowledge that this use may be different from how it may be defined in other work.

a fixed horizon, followed by a MLP head -trained on the raw data b) a single straightforward feed-forward neural network, which uses previous representation learning used in [90] [2]

The underlying assumptions and geometry which we encode in our training scheme are as follows:

- Each septic[3] patient faces mortality risk, and whilst the underlying physiological causes and infections may be different we can still define a form of similarity using the risk a patient faces. Whilst this can be approached using probabilistic methods, we avoid complications in framing the problem in a probabilistic manner by using semi-supervision. In particular, we require a level set of the unit n-hyperball to consist of the equivalence class of all patient states facing the same risk of death.
- As two patients with the same mortality risk can have two fundamentally physiological causes (for example different organ failures), these embeddings should be on the same level sphere, but on different parts of the sphere.

To achieve these goals, we have to project the embedding into the unit closed ball, in contrast to contrastive methods, where the embedding is constrained to the sphere [26, 48]. Further, we do not have a strict disjoint set of classes, so we cannot use any class based losses such as [34, 79]. Instead, in addition to similarity in terms of survival, as we stated above we use a softer notion of similarity such as organ failure, noting that it can be possible for a given patient to have multiple organ failures. We also use a triplet based optimization scheme as opposed to using more recent developments in contrastive representation learning such as [122].

We show several benefits of the proposed method, for both assisting clinicians and for downstream machine learning tasks. For example, the learned embeddings can be used to identify possible new organ failures in advance, and provide early warning signs via the angle of the embeddings and identify increased mortality risk using the norm. The later being considerably better than SOFA score as a predictor for mortality risk for septic patients.

Our work was partially motivated by the desire to introduce a systematic criteria of

[2]This choice was made to be consistent with state definitions used in the RL step-which in turn was chosen to be consistent with previous research using RL for sepsis.

[3]As we mentioned earlier, we illustrate our method on the specific example of septic patients, but the method is readily applicable with minor modifications for any critically ill patient distribution.

defining rewards for offline reinforcement learning (RL) applications in medicine. There has been a lot of interest recently in leveraging RL for critical care applications [63, 99, 77]. However, there are significant challenges at all levels: a most crucial challenge being a lack of an obvious notion of rewards. Some previous applications of RL for sepsis have for example, have used just terminal rewards [63] (i.e. a reward for the final time point of a patient stay depending on release or death) whilst others have used intermediate rewards based on clinical knowledge and organ failure scores [99]. Given the limited number of trajectories and the vast heterogeneity amongst critically ill patients, we hypothesize that terminal rewards do not suffice by themselves to learn the desired policies. Indeed, our experiments show that policies and value functions are qualitatively different and more consistent with medical knowledge when we use intermediate rewards. Research in RL has also shown performance and convergence can be improved when the agent is presented a denser reward signal [69]. Therefore, we show how a reward can be defined systematically using the learned embeddings, and explore the differences in the policies and value distributions. However, we do keep the RL discussion deliberately brief, and defer a further analysis for future work.

In summary, our major contributions are as follows:

- We propose a novel learning framework where high dimensional electronic health record (EHR) data can be encoded in a closed unit ball so that level spheres represent (equivalence classes of) patients with same mortality risk and patients with different physiological causes are embedded in different parts of the sphere.
- We introduce a loss to encode the desired geometry in the unit ball, since the standard losses in metric learning and contrastive learning were ill-suited for this purpose. Further, we describe a simple sampling scheme suited for this method, and show how the sampling scheme and basic domain knowledge can obviate the need to construct data augmentations.
- We experiment using a diverse sepsis patient cohort, and show how the method can identify mortality risk in advance, as well as identify changes in physiological dynamics in advance.
- We show how this learned embedding can be used to systematically define rewards for RL applications. Such a definition changes the value functions and the policies considerably, when compared with using only terminal rewards.

7.1 Related Work

7.1.1 Contrastive Learning & Representation Learning for Clinical Time Series

Self supervised learning and contrastive methods have enjoyed increased popularity and success in recent years, particularly in computer vision and natural language applications [26, 48, 27, 34, 79, 50]. Self-supervised learning methods can categorized into two broad categories [48]. Pretext tasks, where an auxiliary task is solved with the intention of learning a good intermediate representation. Loss function based methods were a representation is learned by directly optimizing an intelligent loss function. We use the latter approach here.

Whilst contrastive representation learning has been popular in other domains, the only similar application to EHR time series we are aware of is [131]. They propose supervised and self-supervised contrastive learning schemes for EHR data, using a neighborhood criteria for the supervised version.

Our work differs from [131] in several ways. First, our method does not require artificial augmentations to define similarity. However we do use very basic medical knowledge about critically ill patients. In that sense, our method belongs to the class of semi-supervised learning rather than self-supervised learning, where most previous contrastive methods were used, with notable exceptions being supervised contrastive learning [60] for images, and some recent work on semi-supervised contrastive learning for automatic speech recognition [129]. This work also significantly differs with respect to the optimization and sampling scheme from all of the previous contrastive methods.

As noted in [131], there have been research on using deep representation learning for EHR data both in isolation [81], and in the context of RL [61, 74, 90]. [81] uses sequence to sequence models in both pretext (forecasting future signals) and loss function (autoencoding) contexts. Denoising stacked autoencoders were used by [86], to create a time invariant representation of patients. Autoencoders were also used by [67], to stratify patient trajectories to a lower dimensional vector.

7.1.2 Reinforcement Learning for Medicine

There has been considerable interest in leveraging RL for medical applications [63, 99, 61, 77, 99, 90, 96]. There have also been guidelines and discussions on challenges associated [45]. However, for the best of our knowledge the only other work which deals with systematically defining rewards is [96]. There, the authors define a class of reward functions for which high-confidence policy improvement is possible. They define a space of reward functions that yield policies that are consistent in performance with the observed data, and the method is general for all Offline RL problems. In comparison our method presented here is a simple by product of the learned embedding and has a simple clinical interpretation for critically ill, where reduced mortality is the primary goal.

7.2 Deep Normed Embeddings: Learning and Optimization

We now motivate our training scheme and optimization criteria, before providing the mathematical formulation. Figure 13 illustrates the geometry we encode on the unit ball, using a 2-dimensional ball as an example. Our optimization algorithm is based on a triplet sampling scheme.

In each triplet, the anchor is a terminal state, either a death state or a release (survival) state. The remaining two states are sampled such that, one is a survivor state and the other is a non-survivor state: both in the last t hours of the corresponding stay. (With t being a hyper-parameter, which should be interpreted as being sufficiently *close* to death or release. We used $t = 12, 24, 48, 72$ in our experiments). The state which has the same outcome as the anchor is labeled as positive, the other is labeled as negative. (For example, if the anchor is a death state, then the non-survivor state is labeled as positive and the survivor state as negative. Note that here, the word positive denotes the similarity to the anchor and not the desirability of the given state.)

The triplet of states is then sent through a neural network parameterized by θ, with $f_\theta(x)$ being the lower dimensional embedding of an input x to the network. The optimization

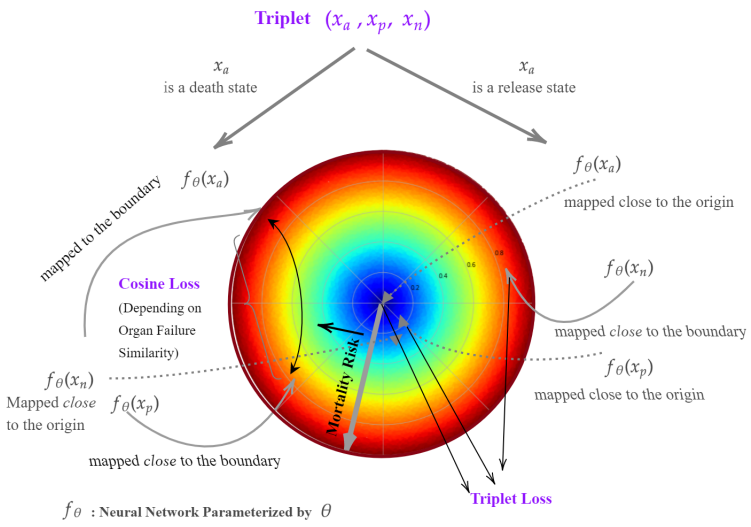


Fig. 13: **The proposed training scheme:** We use a triplet based sampling scheme, where 3 patient states are sampled. One of them, the anchor, is always a terminal state (corresponding to death or release), and the others include a near death and a near release state. Our loss function is then defined in terms of the end result of the anchor state as shown in the figure.

scheme *learns* the neural network parameters such that similar states are mapped to proximity while distance between dissimilar states is maximized, and simultaneously the anchor death states are mapped to the boundary and the anchor release states are mapped to *near* [4] the origin. The positive and negative states, are also mapped near the boundary or the origin, depending their end outcome.

In addition to dissimilarity between survival vs non-survival states, we use an additional level of dissimilarity among non-survival states that occur due to different organ failure modes. Critically ill patients can face mortality risk in various different ways. For example, septic patients display enormous heterogeneity in the underlying infection and the primary organ failure. Therefore, we require our embedding to identify similarity among the patient states using partial supervision. In our example of septic patients, we use four major organ system scores : i) Cardiovascular ii) Central Nervous System (CNS) iii) Liver and iv) Renal, and pick

[4]The releases states should not be mapped exactly to the origin as even survivors have some risk of mortality, and research has shown there is a substantial readmission risk and a shortened life time for septic patients, even when they survive the ICU stay.

the organ system with the worst (highest) score as the worst organ system. Each non-survivor state in the triplet is then annotated with the worse organ system. When the anchor state is a death state, we use a cosine embedding loss, between the two embedded non survivor states. Informally, the goal is to maximize the angle of the embedding of states corresponding to different organ failures and minimizing the angle between two states corresponding to the same organ system failure.

When the anchor is a release state, instead of the cosine embedding loss we use the triplet loss, between anchor, positive and negative. This enables the patient states to be spread across the hyper-ball, and the high mortality risk states to be differentiated from less risky states.

Formally, we optimize the loss function

$$\text{loss}(x;\theta) = \beta(\text{loss}_{\text{terminal}}(x;\theta)) + (1-\beta)(\text{loss}_{\text{contrastive}}(x;\theta)) + \text{loss}_{\text{intermediate}}(x;\theta) \tag{30}$$

Here θ denotes the neural network parameters we are optimizing, and x a triplet of the form (x_a, x_p, x_n) (The implementation uses batches of triplets which is the norm in Deep Learning). The loss function in (30) consists of three components: the terminal (or anchor) loss, the contrastive loss, and the intermediate loss for non terminal states. The first two losses are the most important, and are balanced by a hyper-parameter $\beta \in [0, 1]$.

We now describe each component separately. For ease of notation we will use $d(x)$ for $||f_\theta(x)||_2^2$, where $||x||_2$ denotes the l_2 Euclidean norm on the embedding space. (We use the square of the norm instead of the norm itself purely for the ease of optimization.)

The *terminal loss*,

$$\text{loss}_{\text{terminal}}(x,\theta) = \mathcal{I}_{\{x_a=death\}}((d(x_a)-1)^2) + \lambda_1 \mathcal{I}_{\{x_a=release\}}(d(x_a)) \tag{31}$$

essentially distributes the terminal states to the correct part of the ball. (with respect to the embedded norm). I.e. the death states are embedded on the boundary and the release states near the origin. As we explained previously we want to be more generous on release states mapped away from the origin, since survivors could exhibit non-trivial mortality risk for critically ill patients. Thus we discount the release term with $\lambda \leq 1$ to encourage the network to learn these patterns automatically.

The *contrastive loss*,

$$\text{loss}_{\text{contrastive}}(x,\theta) = \mathcal{I}_{\{x_a=release\}}\text{tripletloss}(x_a,x_p,x_n) + \mathcal{I}_{\{x_a=death\}}\text{cosineloss}(x_a,x_p,y_{ap}) \quad (32)$$

is responsible for determining the separation of states. This loss depends on whether the chosen anchor is a dead state or a release state. Triplet loss is the standard loss as introduced in [109] defined as:

$$\text{tripletloss}(a,p,n) = \max\{||a-p||-||a-n||+\text{margin},0\} \quad (33)$$

We used 0.2 for the triplet loss margin. The cosine embedding loss is only considered when the anchor is a death state. This term depends on the similarity of the two non-survivor states y_{ap}, where $y_{ap}=1$ if both the states belong to the same class and 0 otherwise. We experimented with two options for the cosine embedding loss:

(i) The standard cosine embedding loss used in metric learning defined as :

$$\text{cosineloss}(x_a,x_p,y_{ap}) = \begin{cases} 1-\cos(f_\theta(x_a),f_\theta(x_p)) & y_{ap}=1 \\ \max(0,\cos(f_\theta(x_a),f_\theta(x_p))-\text{margin}) & y_{ap}=0 \end{cases}$$

(ii) Cosine loss based on inner product $<,>$:

$$\text{cosineloss}(x_a,x_p,y_{ap}) = \mathcal{I}_{\{y_{ap}=0\}} < f_\theta(x_a), f_\theta(x_p) >$$

Where $cos(a,b) := < a,b > / \sqrt{< a,a >}\sqrt{< b,b >}$.

Thus, we expect formula (ii) to be similar to (i) near death states, where $\sqrt{< f_\theta(x_a), f_\theta(x_a) >} \approx 1 \approx \sqrt{< f_\theta(x_p), f_\theta(x_p) >}$.[5] Our results in the next sections used the first version with a margin close to 0. Using the second version was more stable in training, but the separation of different organ systems were more clear when the first version was used.

The *intermediate loss* is intended to help the network by mapping near death states near the boundary and near release states near the origin. We note that there are a few

[5]Note that is this formulation we only use similarity as a loss when the organ failures are different. In either case. the anchor state is a death state.

hyperparameters in this loss, but our experiments show that the method is quite robust for most *reasonable* hyperparameter choices.

$$\begin{aligned}\text{loss}_{\text{intermediate}}(x,\theta) = \lambda_2(&\mathcal{I}_{\{d(x_p)>1\}}d(x_p) + \mathcal{I}_{\{d(x_n)>1\}}d(x_n)) \\ &+ \lambda_3(\mathcal{I}_{\{x_a=\text{Death}\}}e^{-\alpha d(x_p)} + \mathcal{I}_{\{x_a=release\}}e^{-\alpha d(x_n)}) \\ &+ \lambda_4(\mathcal{I}_{\{x_a=death\}}d(x_n) + \mathcal{I}_{\{x_a=release\}}d(x_a))\end{aligned} \tag{34}$$

This loss comprises of three components. The first term ensures that the embeddings are constrained to the closed unit ball by penalizing if the squared norm of the embedding is greater than one. We noticed that such an implicit regularization is more effective than explicitly constraining the output of the network. The second and third terms help the learning process, by mapping the intermediate (positive and negative) terms close to the boundary or the origin. We use an exponentially decaying loss for the non-survivor states, so the loss only large, if the norm is close to zero. α is a hyper-parameter which chooses the desired decay. Similarly the last term ensures the near release survivor states are mapped close to the origin in general. However we choose λ_4 to be much smaller than λ_1 and λ_3, so that the network can still identify high risk states.

We discuss the effect of hyper-parameter choices, and present an ablation study in the next section (Results). Further, we note that it is also important to use an orthogonal weight initialization [51] in order to learn a distributed representation on the ball and to prevent dimensionality collapse [54]: This will also be illustrated under Results.

7.3 Results

We will now present some results of our method. The results in this section uses the recurrent neural network architecture. (See supplementary material for implementation details). Some corresponding results for the MLP are presented in the supplementary information.

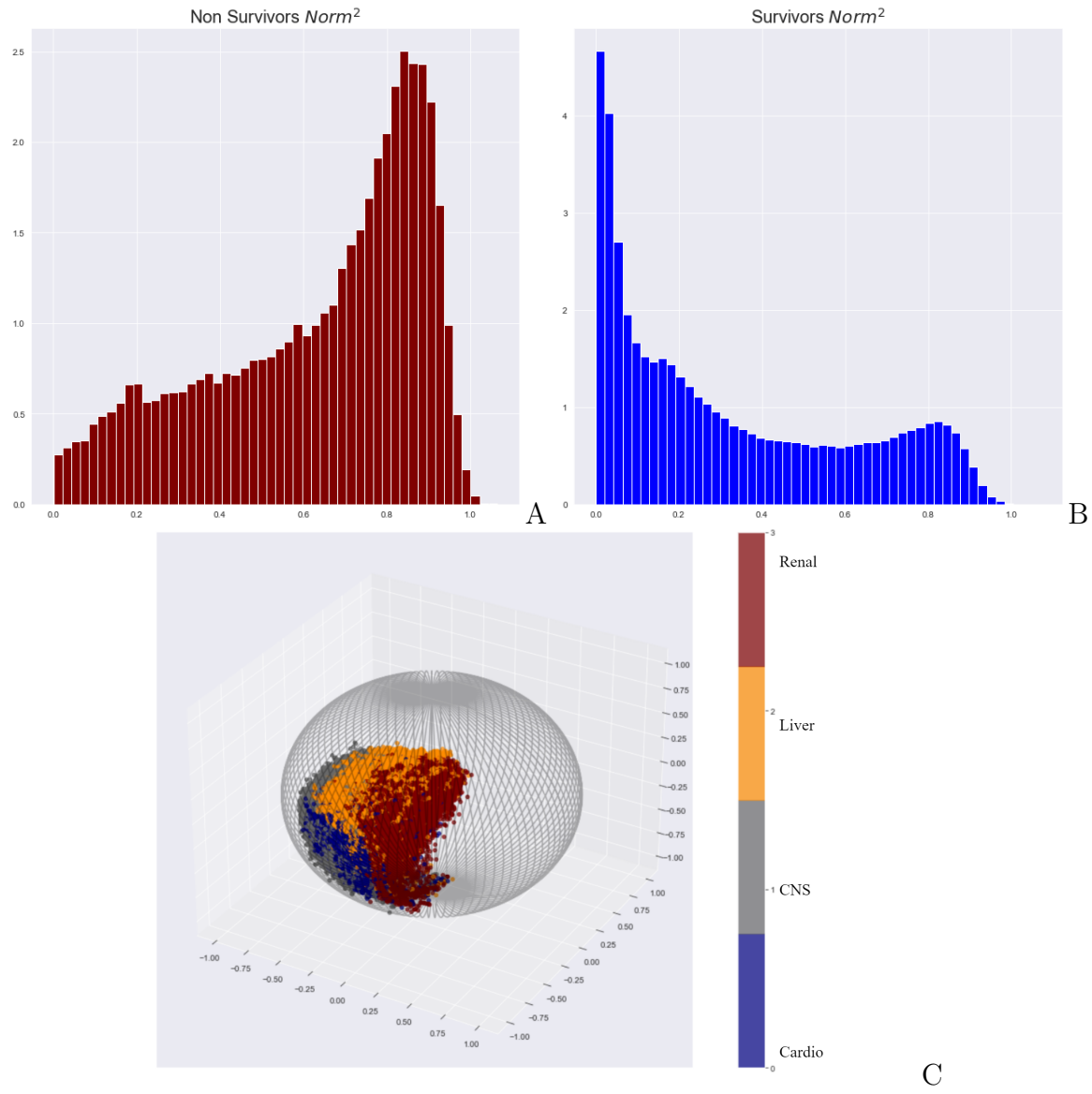


Fig. 14: **A:** Norm^2 of validation cohort non-survivors, **B:** Norm^2 of validation cohort survivors, **C:** A sample of non-survivor patient states, marked by the worst organ system

7.3.1 Patient Representation on the Unit Ball

For ease of visualization, we present results using representations embedded in the 3d unit ball, however, the method works for embedding into any dimension.

Figures 14A and 14B show histograms of squared norms of the embeddings for all survivor

and non survivor patient states (across all time points) in the validation cohort. [6] As the figures clearly demonstrate, the learned embedding associates the norm (or alternatively, the level set $\mathbf{S_k}$ of the form $\mathbf{S_k} = \{x : ||f_\theta(x)||_2 = k\}$) of the embedded vector with mortality risk, with survivor states in general belonging to the lower level sets, and the non-survivor states belonging to the higher level sets. We later show how the norm can be used as an indicator of patient mortality risk, compared with the existing scores such as the SOFA score.

Figure 14C presents a randomly selected sample of patient states, embedded into the 3-dimensional closed unit ball. The colors mark the worst organ system for each state. There is a clear separation amongst different organ failures. We envision, such a presentation can be used to provide real-time visualization to assist clinicians at the ICU. For example, the embedding can be used to identify a patient trajectory heading towards a new organ failure. The embedding being continuous is naturally more granular than the discretized, organ failure scores which were used as an guidance to the network to distinguish different organ failure scores. Indeed, an example of such a patient trajectory is given in Figure 15.

Here, two embedded patient trajectories are plotted in the 3d-unit ball. We focus on the longer trajectory, which is colored in black and green. We focus on the final 50 hrs of this patient's stay. The patient's organ failure scores change at 36 hrs. At this point the patient's the cardiovascular score changes from 4 to 3 and then to 1 at 37 hrs. To show how the embedding *predicts* this change in the underlying physiology before the organ scores reflect it, we color the lines of first 36 hours of the trajectory in black and the last 14 in green. For the first 36 hours the labeled worst organ system is cardiovascular (although, we note for this patient renal and CNS scores were equal to the cardiovascular score.) and hence marked in blue stars. As the cardiovascular score decrease the worst organ system was labeled as CNS and is marked in purple for last 14 hrs. We can notice that the trajectory approaches its final points, even when the organ failure scores do not indicate the increase in cardiovascular scores. Indeed the black lines take the trajectory very close to its end set of points. This is an example of how this learned representation can warn clinicians on changes in patient dynamics. As we can see from this example, the representation can identify these patterns

[6] These results use a network trained with $\beta = 0.75, \lambda_1 = 0.7, \lambda_2 = 10.0, \alpha = 3, \lambda_3 = 0.2, \lambda_4 = 0.05$. Other choices are discussed later.

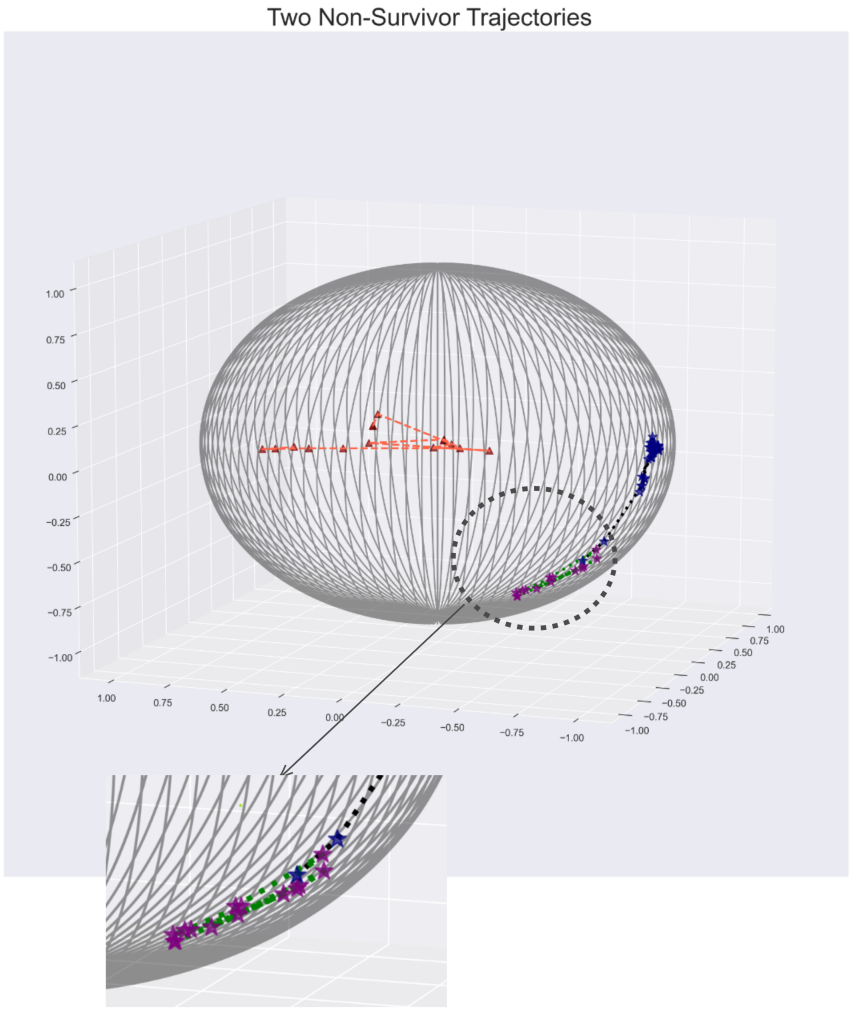


Fig. 15: **Embedded trajectories for two non-survivors:** One patient is labeled with star markers and black/green trajectory, the second with triangle markers and orange trajectory. The marker color indicates the system with the highest organ failure score: Cardio (blue), Liver (Maroon) CNS (Purple). The first trajectory is 50 hrs long, black for the first 36 hrs, green for the last 14. The highest severity organ failure changes from cardio to CNS at 36 hrs. The embedding trajectory approaches the cluster a few hours before the organ scores indicate the change (see detail).

from the data and is not constrained by the supervision signal (in this case the organ failure scores) it was given.

The other trajectory is presented for comparison. This is the final 15 hours of another

non-survivor. As we can see this patient approaches a different part of the boundary as they become closer to death.

7.3.2 Hyper-parameter effects

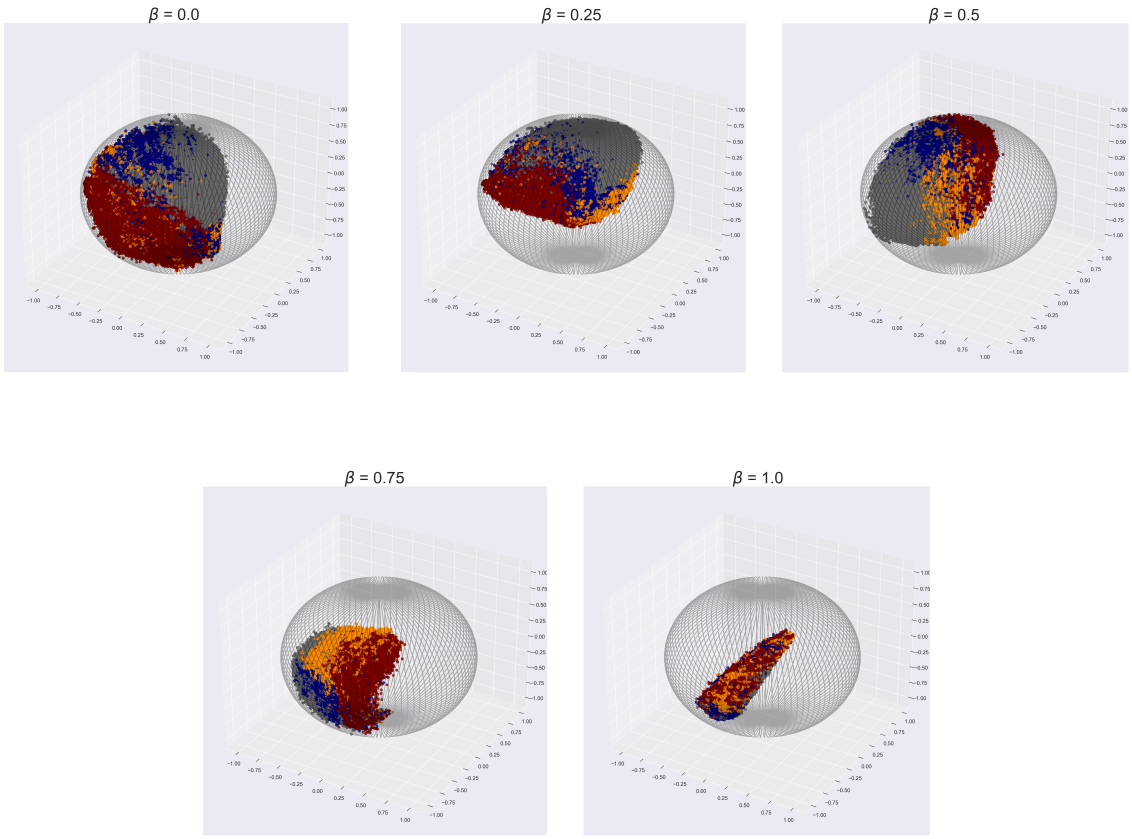


Fig. 16: Embedded state distributions for various β: The labels indicate the worst organ systems as in Figure 14

Now, we will explore the effect of hyper-parameter choices- starting with β. Figure 16 presents the same patient states shown in Figure 16(C) embedded using different β values. Recall that β balances the terminal loss and the contrastive loss. The former focus on determining the correct level sphere and the latter on the angle between states. Thus, Figure 16 is not surprising. The states are spread across a larger portion of the ball as β gets smaller. The figure also illustrates the importance of the contrastive loss, as when $\beta = 1$ all the

states are enclosed into a manifold of much lower volume. The perceptible separation of organ failure modes is also lost. However, encouragingly for all other values of β, there is a separation.

Figure 17 presents the embedded states, when no orthogonal weight initialization was used. As we can observe, the volume of the space covered is much smaller.

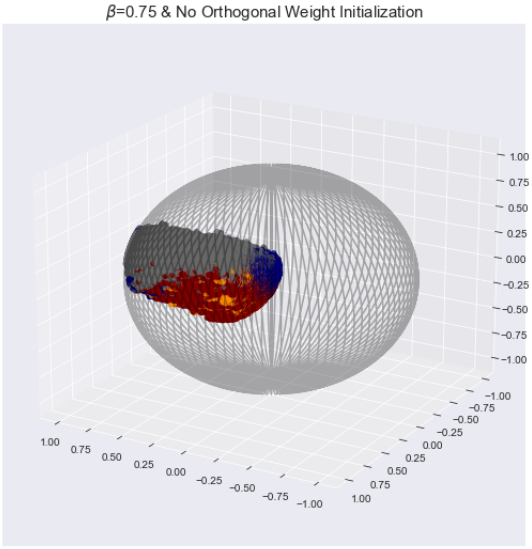


Fig. 17: Embedded state distributions without orthogonal weight initialization. Labels indicate the worst organ system

We also examined the effect of β, on the embedded norm. For this we stratified patient states by: a) survivor and non-survivor, b) times to death or release. Averaged squared norms for different values of β are presented in Figure 18. The observations are as expected: higher β values on average perform better, in mapping states into a more suitable level sphere. The only exception is for survivor norms, where $\beta = 0.75$ has resulted in lower norms than $\beta = 1.0$. Unsurprisingly, when the terminal loss is not used ($\beta = 0$), the norms between the survivors and non-survivors are similar to each other.

In each case, the averaged squared norms increase with time to death and decrease with time to release.

We then followed the same steps for the intermediate loss. These results can be explored in Figure 19. As expected, excluding exponentially decaying (either by setting $\lambda_3 = 0$ or $\alpha = 0$) result in lower norms, on average. This was indeed the motivation for using such

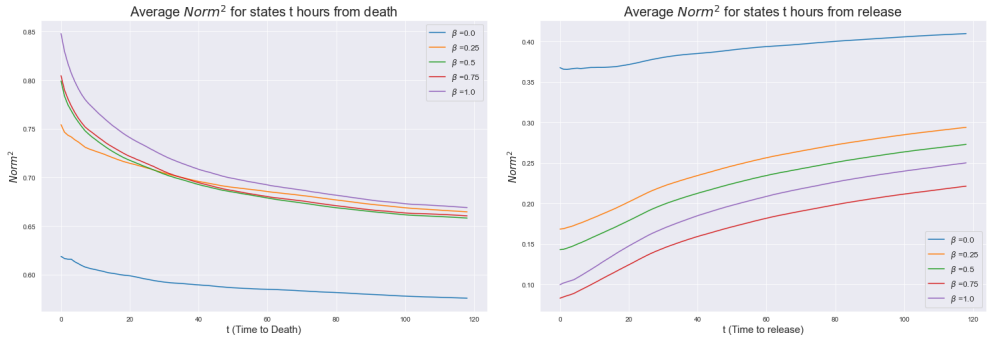


Fig. 18: Averaged embedding norm with time to death (for non-survivors) and release (for survivors): for different β

a term in our loss function. Similarly, when λ_4 is set to 0, on average norms get larger. It is difficult to evaluate the importance of loss components using Figure 19 in isolation. However, intuitively, there seems to be value in both intermediate loss components. Since the intermediate loss only considers the norm and not the angle, it doesn't have any direct impact on the separation of physiological causes (although it implicitly impacts the effect of β).

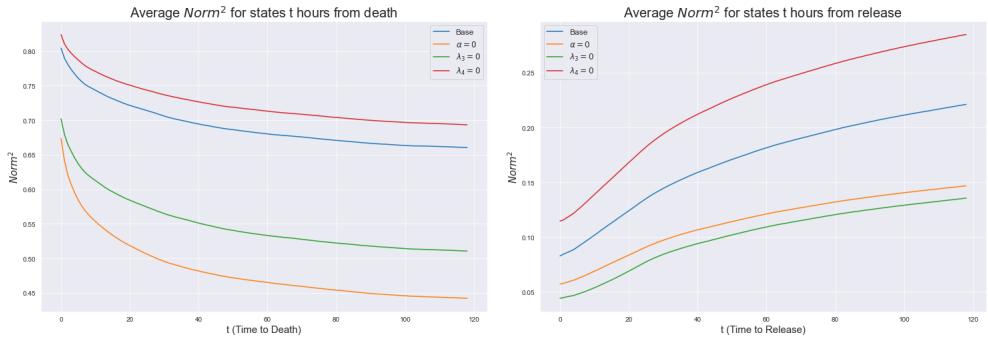


Fig. 19: Averaged embedding norm with time to death (for non-survivors) and release (for survivors): for various intermediate loss choices

We also highly desire some smoothness of the norm trajectory generated by a single patient trajectory. This is particularly important for RL, as we aim to use the difference of the norm (or a monotonic function of the norm) between two consecutive time steps to specify rewards. To quantify this. we computed relative jumps (i.e. $\frac{|d(s_{t+1})-d(s_t)|}{d(s_t)}$). Averaged

results across all patient states are presented in Table 5 for β and 6 for intermediate loss.

Table 5: Averaged relative jumps for various β

β	Average relative jump
0.0	0.128
0.25	0.125
0.5	0.162
0.75	0.264
1.0	0.686

Table 6: Averaged relative jumps for various intermediate loss choices, with $\beta = 0.75$

Choice	Average relative jump
Base	0.264
$\lambda_3 = 0$	0.320
α=0	0.319
$\lambda_4 = 0$	0.160

From Table 5, we can notice that higher β values result in more wiggly curves. We also observed this visually. Again, this phenomena can be explained by the form of the loss function. Higher β values, focus heavily on the terminal states. Therefore, the loss could be minimized by projecting states closer to the boundary or the origin more frequently.

The effects on intermediate loss terms are more interesting. It seems as α (and thus λ_3) has a smoothing effect. However, λ_4 seems to be having a increase the magnitude of the jumps.

7.3.3 Norm as a Predictor of Mortality Risk and Representation Learning for Downstream Machine Learning Tasks

We investigated how the embedded norm can be used as a predictor of mortality risk. For this, we created auxiliary tasks of predicting if a state is within 12, 24, 48, 72, or 120 hours of death. We further used these tasks to compare the quality of the learned embeddings to other representation learning methods. For the latter goal, we used a linear protocol which is common in the common evaluation protocol in computer vision representation methods [48].

First, we calculated the area under the ROC (AUROC), using the norm as the score associated with each state, for each task. For comparison, we followed the same steps with the SOFA score, since SOFA is used as a predictor of mortality for septic patients [38]. The results are presented in Table 7. We can notice that the AUROC with respect to the SOFA score is very similar for each task, therefore we also computed the AUROC using a SOFA type, the aggregate score of just 4 organ systems: cardiovascular, CNS, liver, and renal. [7].

Table 7: AUROC for predicting if a state is t hours from death for various t

Task (t)	SOFA	SOFA(4)	Norm ($\beta = 0.5$)	Norm ($\beta = 0.75$)
12 hrs	0.717	0.746	0.847	0.836
24 hrs	0.716	0.741	0.828	0.8176
48 hrs	0.715	0.731	0.807	0.798
72 hrs	0.715	0.725	0.797	0.790
120 hrs	0.719	0.727	0.789	0.782

We do note that the SOFA score is an aggregation of different organ failure scores, and a patient can face mortality risk from just a few organ failures. Therefore it is not a perfect score to measure mortality risk. However, it is still used regularly at the ICU to predict mortality risk and it is encouraging that the learned embedding has shown a significant improvement in AUROC. The benefit of our method is that it can indicate the risk *and* the organ failures (or physiological causes in general) responsible. This would not have been possible if the method was approached from probabilistic methods for example.

However, we used this problem to investigate the quality of the learned representation, against other standard representation learning methods. For this, we used a linear evaluation protocol by simply fitting a logistic regression model on top of the learned representations. We did this on 100 different train, test splits: training a logistic regression model on one and noting the test AUROC.

For comparison, we learned a recurrent denoising autoencoder [125] with the same architecture. Briefly, denoising autoencoders attempt to learn an intelligent representation by reconstructing a corrupted input by a) first projecting into a lower-dimensional space and then b) decoding this embedding. This method is similar to the autoencoding method

[7] Whereas the full SOFA score uses 6 organ systems

used in [81] [8]. As mentioned under related work, autoencoders are a popular choice for EHR representation learning [86, 67]. We noticed that our method significantly outperformed the denoising autoencoder of the same hidden dimension, and thus we also used a denoising autoencoder of a four times larger hidden dimension (12). Further, we used a standard triplet learning scheme. Here, a randomly selected patient state is corrupted by injecting noise to define a positive state. A different patient state belonging to a patient with the opposite end outcome is then selected as the negative state. Then triplet contrastive loss is used as the objective. More implementation details and problem specifications of these methods are included in the supplementary material.

Finally, we also fitted a logistic regression model on the full observations. The results are presented in Table 8. For the normed embeddings we show results corresponding to $\beta = 0.5$ (including a) just the 3d embedding b) embedding and the norm as another feature) which performed the best-numbers corresponding to other choices are stated later. All the models were trained on the same train, test splits.

Table 8: Average test AUROC for predicting if a state is t hours from death for various t

Task (t)	Full Observed	Emb.	Emb. + Norm	Denoise Auto (3d)	Denoise Auto (12d)	Triplet
12 hrs	0.870	0.863	0.851	0.746	0.834	0.690
24 hrs	0.847	0.837	0.829	0.726	0.811	0.681
48 hrs	0.822	0.812	0.803	0.712	0.787	0.674
72 hrs	0.8097	0.802	0.795	0.704	0.778	0.674
120 hrs	0.796	0.794	0.786	0.705	0.773	0.683

As the results indicate, our method significantly outperforms both baselines of the same hidden dimension. In fact, even after increasing the hidden dimension of the next best alternative, its performance was inferior to the method introduced in this work. Indeed, the AUROC of all tasks are quite close to the AUROCs of models trained on the full 27 dimensional raw input. Further, by comparing both Table 7 and Table 8 we can notice that even the one dimensional norm itself is competitive as a risk score even against a model trained on the full input space *specifically* for these tasks.

[8] However, we use the same simple architecture as above, instead of the attention based architecture used in that work.

7.3.3.1 Ablation: β and Intermediate Loss

Next, we will present the AUROC statistics for various β values and intermediate loss choices. We present results of fitting logistic regression models across 100 different train and test splits. In addition, we also computed the AUROC when the norm is used as the score. For simplicity, we averaged over all the above tasks. The results are presented in Table 9 for β and 10 for the intermediate loss. [9] It is interesting that with respect to this task $\beta = 0.25, 0.5$ have superior numbers despite higher β values focus more on the embedded norm. Even the model trained with no terminal loss, (recall the intermediate loss is still used) performs reasonably.

Table 9: Average AUROCs for different β

β	LR AUROC	Norm AUROC
0.0	0.783	0.693
0.25	0.824	0.815
0.5	0.827	0.813
0.75	0.818	0.805
1.0	0.805	0.803

As Table 10 suggests not using the exponentially decaying terms (either by setting $\alpha = 0$ or $\lambda_3 = 0$), reduces the AUROC. This observation is consistent the previous ablations (Figure 19. The effect of λ_4 is however unclear. The norm AUROC reduces, when $\lambda_4 = 0$. However, the logistic regression AUROC improves slightly.

Table 10: Average AUROCs for intermediate loss choices- $\beta = 0.75$ in each.

Choice	LR AUROC	Norm AUROC
Base	0.822	0.805
$\lambda_3 = 0$	0.810	0.800
$\alpha = 0$	0.795	0.787
$\lambda_4 = 0$	0.828	0.797

We emphasize that whilst these results are promising, these tasks are artificial. Therefore, performance with respect to the AUROC by itself is certainly not enough to claim that

[9] Notice that the two tables were generated independently. All the models in the same table used the same train-test splits, however the splits were different across the two tables. Thus, the numbers corresponding to $\beta = 0.75$ in Table 9 and base in Table 9 are different.

our representation learning method is necessarily superior to other approaches or that a specific hyper parameter combination is superior. Benchmark tasks are popular amongst various machine learning communities. However, evaluating medical machine learning methods (especially unsupervised, representation learning methods) using adhoc tasks can be ineffective and even dangerous. Thus, we intentionally avoided conducting a large number of arbitrary experiments. However, the method introduced here is flexible to be adapted to most similar medical machine learning tasks. Further, it presents enough opportunities to encode domain knowledge.

7.3.4 Reinforcement Learning: Rewards and Representation

In this section, we discuss how the learned embeddings can be leveraged for RL. For consistency between RL state spaces and the inputs of the representation learning, the results of this section uses the MLP architecture. In particular, both methods takes the same *state* as input. We present a detailed description of the RL methods and implementation details in the supplementary material.

To be consistent with previous work [90], we use deep distributional reinforcement learning using the categorical c51 algorithm [9], which approximates the return distribution with a discrete distribution with fixed support. The state and action spaces are also identical to that work (except when using the embedded vector for state augmentation). We keep the RL methods simple. For example for the results presented here, we do not re-weight the patient distribution when sampling as in [90], and we assume actions are taken with respect to the expected value of each value distribution.

As mentioned previously, our intention is purely to illustrate how the proposed low-dimensional embedding can be used to define rewards, aid in state augmentation, and how such a choice affects the recommended policies. Evaluating RL agents in the offline setting is an open problem and an active research area, and the current off-policy evaluations (OPE) are particularly ill-suited for critical care applications [46]. Even when OPE methods can be used they are defined in terms of a fixed reward specification, making it impossible to use them for comparison of RL algorithms learned under different reward functions. Therefore,

we do not claim the methods proposed here are superior than the existing methods for RL. However, our results show qualitative differences in values and policies that meet clinical intuition, hinting towards the benefit of this formulation.

We experimented with two formulations of intermediate rewards. In each case we used terminal rewards of -15 for terminal death states. For terminal survivor states, (release states) we use $15(1-d(s))$ as the terminal reward. This was done to acknowledge that not all survivors are the same and there could be patients with higher mortality risk even amongst survivors. Indeed medical research have claimed that the life expectancy reduces significantly even for sepsis survivors. [32, 47]. The scale of 15 was chosen to be consistent with previous work, for example [99].

In our first formulation we define, intermediate rewards of the form:

$$r_1(s, a, s') = 0.375(d(s) - d(s')) \tag{35}$$

[10] where s, a are the current state and action and s' is the next state. Here we use $d(s)$ to denote the square of the norm of the embedded vector of the state s. (i.e. $(||f_\theta(s)||)^2$). We have used the current notation for simplicity, noting the slight abuse of notation. This choice has a natural interpretation of minimizing the cumulative increases of risks between consecutive time steps. However, we noticed (by comparing the outputs of bootstrapped networks) that the variance of the learned norm can be high, so $(d(s) - d(s'))$ can only be considered as a noisy estimate of the difference in risk. However, using the boostrapped networks, it is straightforward to include a form of confidence in this estimate, and then consider a regularized reward to reflect parametric uncertainty. We do not do that here do keep our RL presentation brief.

In our second formulation, we defined intermediate rewards using the norm in the same spirit as how SOFA score was used as an intermediate reward in previous work such as [99]. More specifically, in that work there were two components of intermediate rewards depending on the next state's SOFA score : (i) A change in SOFA score (SOFA score increasing resulting in a negative reward, and decreasing a positive reward) (ii) A negative reward for when the

[10] More generally we can define intermediate rewards of the form $r_2(s, a, s') = \alpha(d(s) - d(s'))$ where $\alpha > 0$

SOFA score does not improve. Further, a 15 or -15 terminal reward was given for release or death, respectively.

Therefore, we define intermediate rewards of the form:

$$r_2(s, a, s') = 3.75(d(s) - d(s')) - 0.25\mathcal{I}_{\{d(s')>0.5\}}d(s') \tag{36}$$

where s, a are the current state and action and s' is the next state. Here we use $d(s)$ to denote the square of the norm of the embedded vector of the state s. (i.e. $(||f_\theta(s)||)^2$). We have used the current notation for simplicity, noting the slight abuse of notation.

Notice that in expression of r_2 the first term is positive if and only if the norm of the next embedded state is less than the current norm. The second term is a penalty included to discourage keeping a patient at a risky state. For our RL experiments, we used a 10d embedding, and further for the norm calculation we averaged the norms of 10 bootstrapped networks. Both of these choices, were intended to reduce the variance of the estimate. Further, we experimented with augmenting the state representation, with the embedded vector.

Now, we will discuss the changes in the policies. We noticed that when we only use terminal rewards, the percentage of states with no recommended treatment is much higher than with intermediate rewards. This phenomena has been observed in previous research [63]. We present a summary of these results in Table 11. Since, there is variability among the treatment recommendations, we present averaged results. The averaging was done using different versions of the function approximating neural network : 5 networks learned independently using bootstrapped patients and weights of the last 3 epochs when the network was trained on the whole training dataset. In addition to averaging the actions recommended by each, we also present results where we average the value functions first and then recommend actions according to the averaged value function. In each, case we can notice that using terminal rewards only causes the action with no recommended treatment more frequently. The breakdown of the full treatment percentages under all 3 reward formulations are presented in the supplementary information.

We will discuss some properties of the value distributions and present the full action distribution in the supplementary material.

Table 11: Percentages of states with no treatment

Method	% states with no treatment
Clinician	27.78
Terminal Rewards-Averaged Actions	52.68
Int. Rewards 1 (r_1)- Averaged Actions	18.33
Int. Rewards 2 (r_2)- Averaged Actions	28.61
Terminal Rewards-Averaged Value Functions	65.00
Int. Rewards 1 (r_1) -Averaged Value Functions	26.40
Int. Rewards 2 (r_2) -Averaged Value Functions	32.16

7.4 Discussions and Conclusions

In this work, we introduced a novel contrastive representation learning scheme suitable for EHR data. One of the key differences between our method and other constrastive methods, across all application domains, is that our method works in the semi-supervised setting rather than purely self-supervised setting. We believe self-supervision using augmentations could be challenging for medical time series, and unfortunately most state of the art constrastive methods depend on heavy augmentations. However, there is enough regularity and domain knowledge which can be exploited, although we do not have strict classes as for example the image domain. Hence, we had to work in the semi-supervised setting rather than a fully supervised setting. Indeed, one of our main aims of this work was to show how minimal and loosely defined supervision and benefit in contrastive learning for clinical applications, and we expect this work to be adapted to reflect different goals in machine learning applications for healthcare.

We have shown that our method has learned to identify mortality risk and changes in patient dynamics in advance in terms of the underlying physiology (via organ failure). We believe such an application can strongly supplement human clinicians at the ICU. The supervision given for this work is minimal and stronger supervision signals about the underlying physiological mechanisms could result in a better and a more interpretable representation. However, this would require more granular data than what is routinely collected at the ICU.

Indeed, one of the key challenges in medical time series is that we do not have access to the same quantity or the quality of data as for example natural language.

There has been recent interest in exploring the geometry of deep learning [19]. In this work, we use simple geometrical priors using the norm and inner products of a lower dimensional hyper-ball to encode the desired behavior. However, in future work, we plan to explore ways of using stronger geometric priors to encode medical knowledge. We believe such a scheme could also improve interpretability of the representations, as well as improve performance of various machine learning tasks. It is also a potential way to leverage well established mathematical theories of differential geometry and topology (amongst others). However, such a use is far from trivial and would require more research.

We also note that, our method could be improved for task-specific applications through hyperparameter optimization and using different neural network architectures. Our aim was to emphasize on the method and the associated geometric intuitions, and thus we did not focus on finding the *optimal* hyper-parameters. Similarly, we note that the performance could be improved by using recent advances in contrastive learning such as what is introduced in [26, 48].

Finally, we showed how the learned embedding can be used to define rewards for RL and how as a result the distribution of values and the policies change considerably. Whilst we have only used the norm of the embedding for RL results presented here, we anticipate this method can be used in other ways for RL and control. For example, the organ system changes can be considered if we can define rewards in terms of the inner product of two consecutive vectors. However it is not immediate how this should be done, so we defer this to future work. We may also interpret the lower dimensional embedding as an action induced patient trajectory and a simplified dynamic patient model (where the action conditioned dynamics will have to be estimated). This should allow us to use model based control methods, and the low dimension could enable us to use more traditional control methods. However this too, would require more research and is another direction we want to explore. Unfortunately, *all* RL methods in medicine are subjected to challenges at all levels, including evaluation. Therefore, we do not make any claims about the performance of the learned RL policies, rather we emphasize the method and how it can be used to set up the RL framework, more

systematically compared to previous work.

7.5 Broader Impact Concerns

We emphasize our aim of this work, is to introduce a novel, *potentially* impactful computational approach. However, as with all computational and data driven approaches to medicine, significant human evaluation is necessary before such approaches can be utilized at the ICU. Thus, we certainly don't recommend this method for practical deployment at its current state.

7.6 Supplementary Information

7.6.1 Data Sources and Preprocessing

We used a fixed cohort for all our experiments. This cohort consisted of adult patients ($\geq$ 17) who satisfied the Sepsis 3 criteria [55] from the Multi-parameter Intelligent Monitoring in Intensive Care (MIMIC-III v1.4) database [56, 93]. The excluded patients included patients with more than 25% missing values (of vitals and scores) after creating hourly trajectories, patients with no weight measurements recorded and patients discharged from the ICU but ended up dying a few days or weeks later at the hospital.

We used already pivoted, hourly vitals and scores available through the MIMIC-project. However, labs were measured more infrequently-in most cases once in every 8-12 hours. Therefore the lab values were imputed using a last value carried forward scheme, with the interpretation that the recorded data is the last measured lab. For both vitals and scores, missing values using a last value carried forward scheme.

More specifically the state consisted of :

- **Demographics**: Age, Gender, Weight.
- **Vitals**: Heart Rate, Systolic Blood Pressure, Diastolic Blood Pressure, Mean Arterial Blood Pressure, Temperature, SpO2, Respiratory Rate.
- **Scores:** 24 hour based scores of, SOFA, Liver, Renal, CNS, Cardiovascular
- **Labs:** Anion Gap, Bicarbonate, Creatinine, Chloride, Glucose, Hematocrit, Hemoglobin, Platelet, Potassium, Sodium, BUN, WBC.

For RL and for the MLP based representation learning we also used the representation learning used in [90]. These states included 4 cardiovascular states and a 10 dimensional lab history representation.

For RL, we used the same action definitions as [90]. For fluids, this was the total hourly volume of fluids (adjusted for tonicity). However for vasopressors it was the maximum norepinephrine equivalent hourly dose (mcg/kg). The vasopressor 1/2 cut off was 0.15 mcg/kg/min norepinephrine equivalent rate. The corresponding cutoff for fluids was 500 ml for fluids. Action 0 denotes no treatment.

In summary, the markov decision process (MDP) used for RL is:

- **State**: The 41-dimensional state space described above
- **Actions**: A 9 dimensional discrete action space, where vasopressors and fluids can take values $0, 1, 2$. 0 indicates that treatment wasn't administrated.
- **Rewards**: Several choices were used (see main text).
- **Time Step:** 1 hr

7.6.2 Reinforcement Learning

In this section, we will briefly mention some RL background.

RL is a framework for optimizing sequential decision making. RL can be formalized using a Markov Decision Process (MDP), consisting of a 5-tuple (S,A,r,γ,p). This includes state and action spaces $\mathcal{S}, \mathcal{A}$, a (typically unknown) Markov probability kernel $p(|s, a)$, which gives the dynamics of the next state, given the current state and the action and a reward process with a kernel $r(|s, a)$. A policy π is a possibly random mapping from states to actions.

Given a discount factor γ, the return is defined as the cumulative discounted rewards : $\sum_{t=1}^{\infty} \gamma^t r_t$, which is a random variable. The objective of an RL agent is to optimize some functional of the return, usually its expected value (Induced by a policy and environment dynamics).

Thus, the *value function*, $V^\pi(s) = \mathbb{E}_{p,\pi}[\Sigma_t \gamma^t r_t(s_t, a_t)|s_0 = s, \pi], \forall s \in S$, is defined as the expected future discounted rewards when following policy π and starting from the state s. The *Q-function*, $Q^\pi(s, a) = \mathbb{E}_{p,\pi}[\Sigma_t \gamma^t r_t(s_t, a_t)|s_0 = s, \pi, a_0 = a], \forall s \in S, a \in A$, which returns the expected future reward when choosing action a in state s, and then following policy π.

Distributional RL methods, attempt to learn the entire probability distribution of the return, rather than focusing on the expected value. Therefore distributional methods can be used to define actions with respect to criteria different from the expected value.

7.6.3 Implementation Details

7.6.3.1 Contrastive Representation Learning

For the autoregressive model, we only considered states after at least 12 hours from admission. For these states, we first send the past 12 hr history (up to and including the current time), through a GRU based recurrent neural network. Then, we concatenated the current GRU hidden state with the current observations and send the new input through a MLP head. The final layer is sent through a tanh non-linearity.

We trained all our networks for just 10 epochs (passes through the training data). In each case, we monitored a validation loss (with respect to the same loss that is optimized) and saved the weights of the network, corresponding to the minimum validation loss.

We used standard, mini-batch stochastic gradient based optimization using Adam [62] with a batch size of 128 and a learning rate of 3×10^{-5}. In sampling batches, we first sampled a number of patients equal to the batch size and their respective terminal states were taken as the anchor states. Then for each patient, a non-survivor state and a survivor state (in the last t hours) from two different patients were drawn randomly and depending on the end outcome of the anchor state, these states were labeled as positive or negative. The worst organ scores corresponding to each state, were also noted.

We further used a weighted sampling scheme, where non-survivors were sampled more frequently (as the anchor). However, this was purely due to the heavily imbalanced nature of our cohort where around 90% of the patients were survivors.

The following table lists all the hyper-parameters used in our implementation. Note that we have mentioned β

For MLP, the same contrastive loss hyper-parameters were used. However, we used larger batch sizes of size 256, It had 12 hidden layers of 512 hidden units and ELU non-linearities. The optimization details were the same as above.

Table 12: Contrastive learning hyper-parameters

Hyper-Parameter	Value
RNN layers	2
MLP layers	8
RNN hidden dimension	128
MLP hidden dimension	512
MLP activation functions	ELU
Weight Initialization	Orthogonal
Optimizer	Adam
Learning rate	3×10^{-5}
Batch size	128
Non-Survivor Sampling weight	5
α	3
λ_1	0.7
λ_2	10
λ_3	0.2
λ_4	0.05

7.6.3.2 Baseline Representation Learning

Denoising Autoencoder: We used the architecture described above as our encoding neural network. However, we did not use the tanh non-linearity at the end. During, training we injected noise by randomly zeroing out entries with a probability of 0.1. The decoder was a MLP with one hidden layer of 128 dimension and a ELU non-linearity. We used a batch size of 128 and Adam as the optimizer with a learning rate of 3×10^{-5}. This network was trained for 25 full epochs, and we used the weights of the network with the best test loss (computed with corruption).

Triplet Contrastive Learning: For the triplet contrastive method, we again used the same architecture, except for the tanh non-linearity. We normalized the output so that all outputs are unit vectors.

We trained by first randomly selecting a patient, and then a state. This was the anchor. We then, injected independent Gaussian noise to each dimension to create a positive version. Next, we sampled a patient that had a different terminal outcome. A random time point of this second patient was taken as the negative state. We again, used batch sizes of 128 and

Adam with the same learning rate. The triplet loss margin was 0.2

7.6.3.3 Auxiliary Tasks

For logistic regression models, we created 100 different train and test splits by first, sampling 80% of patients out of the total cohort, and then taking the data-points of these patients as the training data and the rest as test data. For a given set of features, we trained a logistic regression model on each of the training data and evaluated the test AUROC.

7.6.3.4 Reinforcement Learning

For RL, we used the c51 distributional algorithm [9], with a 51 dimensional support, using batch sizes of 100 and Adam as the optimizer. For bootstrapped networks, we first generated a random number between k 0.6 and 0.85, and selected a random k% sample of patients. Then, the network was trained on these patient trajectories.

Other relevant hyper-parameters are noted in the following table.

Table 13: RL algorithm hyper-parameters

Hyper-Parameter	Value
Support size	51
Maximum value	18
Minimum value	-18
γ	0.999
Batch size	100
Optimizer	Adam
Learning rate	3×10^{-4}
τ	0.005

7.6.4 More Results

In this section, we briefly present results when the MLP architecture was used. However, note that these results were not generated using the same patient cohort as 14.

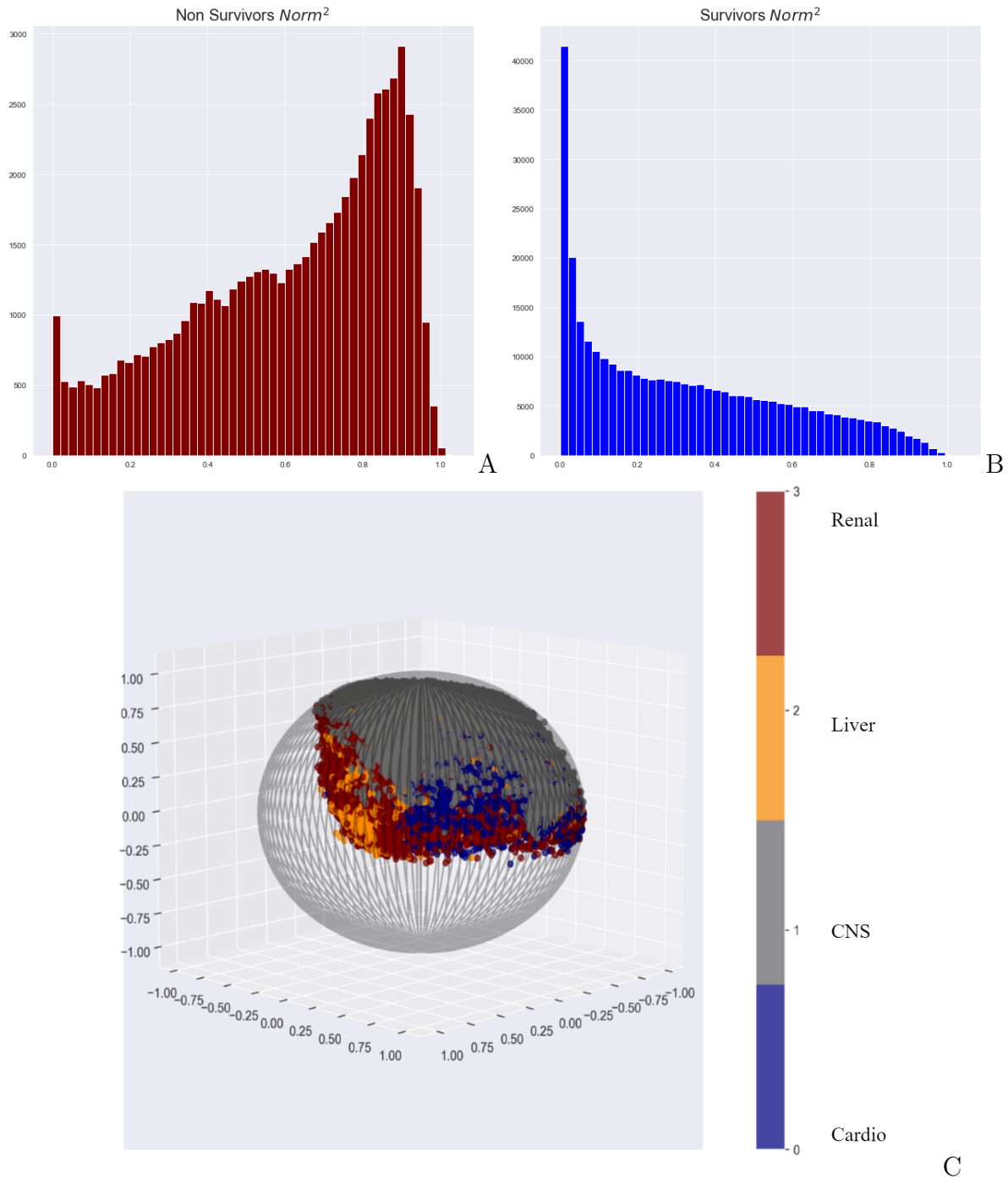


Fig. 20: **Results of the MLP model A:** Norm^2 of validation cohort non-survivors, **B:** Norm^2 of validation cohort survivors, **C:** A sample of non-survivor patient states, marked by the worst organ system

7.6.5 More RL & Control: Results and Discussions

In this section, we briefly discuss some additional results of leveraging our method for RL and control.

For better comparison, we present a table (Table 14) with the percentages of all actions across the whole cohort under the 3 reward schemes. The results presented here are derived from the averaged value distributions, using bootstrapped ensembles.

Table 14: Percentage of recommended actions under different schemes and the clinician

Action	Terminal Rewards.	Int. Rewards 1 (r_1).	Int. Rewards 2 (r_2)	Clinician
Vaso 0 Fluids 0	65	26.4	32.2	27.8
Vaso 0 Fluids 1	19	11.7	0.03	23.7
Vaso 0 Fluids 2	9	18.3	58.2	31.8
Vaso 1 Fluids 0	2.8	7.6	0.9	1.2
Vaso 1 Fluids 1	3.3	23.5	1.1	3.2
Vaso 1 Fluids 2	0	0.2	0.1	4.0
Vaso 1 Fluids 0	0.04	12	7	1.2
Vaso 2 Fluids 1	0.02	0.03	0.2	2.5
Vaso 2 Fluids 2	0.02	0	0.01	4.4

We note that there is a considerable difference between recommendations among the reward schemes. Evaluating between different policies using historical data is one of the hardest challenges faced by any application of RL or control to medicine. Therefore, we don't claim any specific scheme is necessarily better at this point.

However as we have mentioned previously the first formulation does have a natural meaning for critically ill patients, and its increased vasopressor recommendation is consistent with previous RL work for sepsis [90], and recent medical research [112]. We suspect the reasons for the second reward choice to recommend less vasopressors could be that the clinicians usually prescribe vasopressors for high risk patients, thus there are less high risk patient states with no vasopressors administered in our observed data. (r_1 penalizes staying at a high risk state by $-0.25d(s')$) This could potentially be addressed by using offline RL methods for minimizing the effects of distribution shift, but such efforts are differed for future work.

Now, we will compare the optimal values under different formulations.

We will present our results using three different formulations: (i) using only terminal rewards, (ii) using only terminal rewards but augmenting the state with the embedded vector, (iii) using intermediate rewards (No embedded state augmentation). Each was trained using the same hyper-parameters for 8 epochs.

Since the value itself is defined in terms of the reward choice, we scaled all the values using a minimum, maximum scaling scheme, so that for each formulation the values fall in the interval $[0, 1]$. We then, explored the differences of values amongst survivors and non-survivors, expecting a noticeable difference at least when the states are close (in time) to their eventual final outcome.

Due to the more pronounced difference, we will present results which use r_2, first.

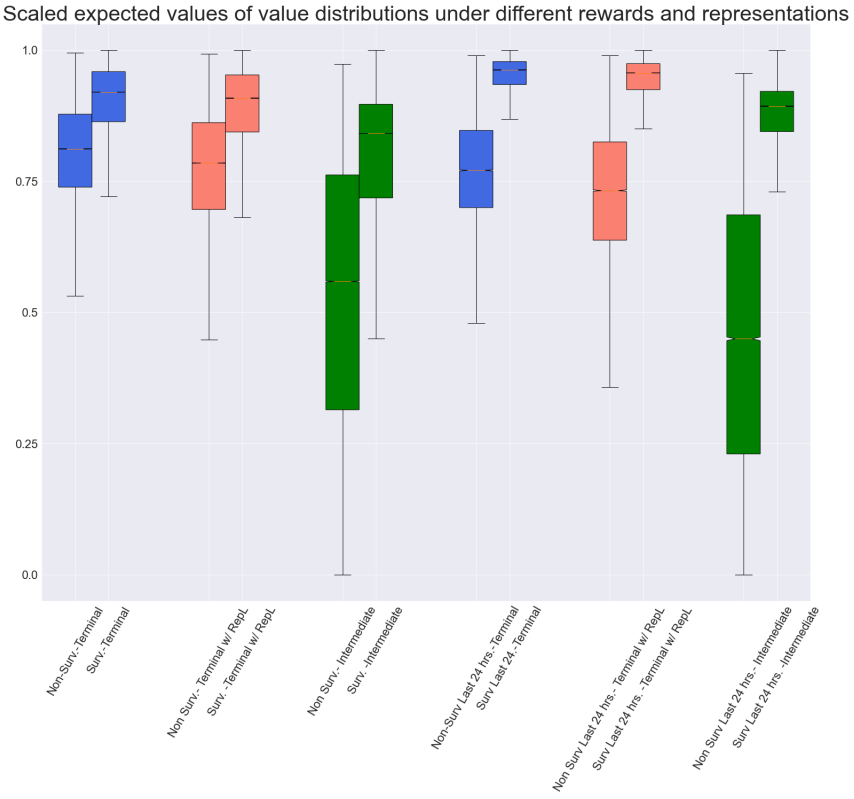


Fig. 21: **Box plots of optimal values:** The results are shown for different reward schemes and representations.

Figure 21 presents box plots of scaled optimal values of patient states. In this figure the intermediate rewards use the formulation r_2. An analogous figure, with r_2 can be found in the supplementary material. For each, we present the box plots for all survivor states, all

non-survivor states, survivors states within 24 hours of release, and non-survivor states within 24 hours of death. It is interesting to note that, the differences in values are most perceptible when intermediate rewards are used. This makes more sense clinically, than results when only terminal rewards were used, where the median non-survivor values are high even when they are 24 hours from death. Moreover, there seem to be a slight increase in difference between survivor and non-survivor quartiles, when the representation learning is used. This is especially noticeable in the last 24 hours of each set of patients.

Figure 22 presents box plots for optimal values for all 3 reward choices. We can notice that when r_1 is used instead of r_2 the differences between survivor and non-survivor values are less pronounced. However, there are still interesting differences when compared with terminal rewards. For example, variance and interquartile range of survivors are much higher. (Recall that the values are scaled using a min-max scheme) In addition, the values of survivors are no longer concentrated near 1.

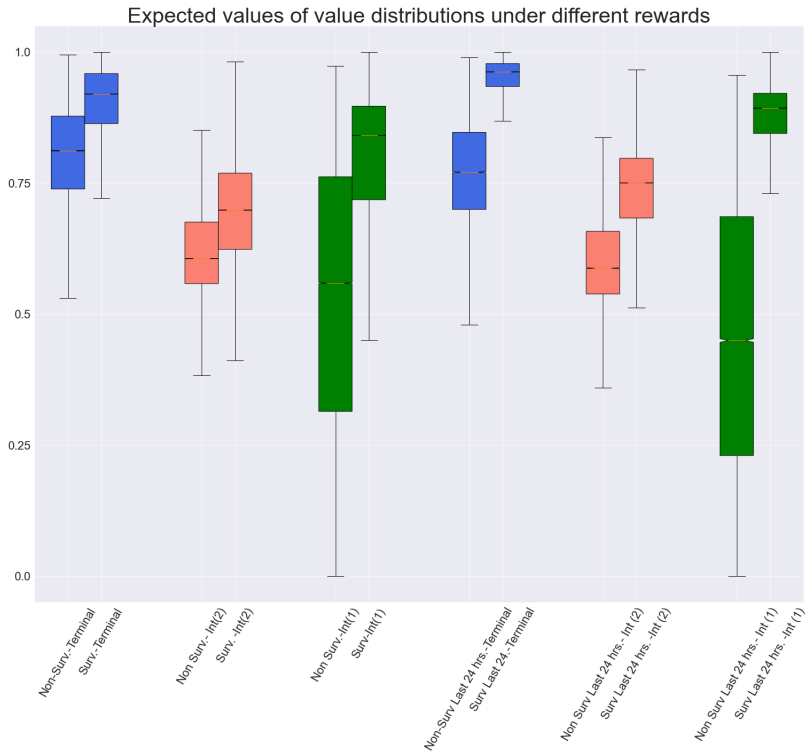


Fig. 22: **Box plots of optimal values:** The results are shown for different reward schemes

8.0 Prologue to Article 3

The next article proposes a clinically motivated control objective for critically ill patients, for which the *value functions* have a simple medical interpretation. Further, we present theoretical results and adapt our method to a practical Deep RL algorithm, which can be used alongside any value based Deep RL method.

We call this method *RL4S: Reinforcement Learning for Survival*. The motivation for this work follows from the ambiguity of quantifying the control objective and rewards for RL. As we discussed in Article 0, rewards are fundamental to RL, and the only way to guide the desired behavior, but it's unclear how they should be defined, in such a way that i) the objective of maximizing discounted cumulative rewards is a reasonable clinical goal and ii) its sample complexity is low.

The abstract control problem motivated us to look at the problem differently. Thus, we approach the problem in a different way starting from a straightforward objective: How to maximize the probability of survival?. We then refine this objective to a practical Deep RL method. We show that there are also alternate interpretations of our methods, including a method that penalizes *unlikely* survivors.

The similarity to DQN type algorithms, allows us to trivially incorporate a range of modifications including : Distributional RL, Risk Sensitive RL, Offline RL and Actor critic methods.

We also perform empirical experiments and show that this method produces clinically intuitive values and seem to discriminate between survivors and non-survivors.

This article will be presented at **Workshop on Interpretable ML in Healthcare at International Conference on Machine Learning (ICML)**. We aim to publish an extended follow up article in the future.

9.0 Article 3: Reinforcement Learning For Survival: A Clinically Motivated Method For Critically Ill Patients

Recently, there has been an increased volume of research which try to learn optimal treatment strategies for critically ill and in particular for septic patients [63, 28, 99, 74, 92, 39, 90], using Reinforcement Learning (RL) methods. Given the enormous mortality, morbidity and economic burden [78, 102, 91], the ambiguity regarding optimal treatment strategies and lack of accepted guidelines for treatment [84, 53], such attempts are certainly justified.

In this work, we will focus on applications where reduced mortality is the primary clinical goal. For such problems, there has been debate on optimal reward choices for the RL formulation. Indeed, some work have used exclusively terminal rewards (for example, $+/-1$ depending on death or release or just a negative reward for death) [63, 74, 61], whilst others have used clinically motivated intermediate rewards [99, 92, 90]. Whilst just using terminal rewards does make sense as a clinical objective, such sparse reward choices induce high sample complexity, and all RL applications to medicine are performed in an *offline* manner, using a fixed dataset of observed trajectories. In particular, for complex syndromes such as sepsis, given the enormous heterogeneity and complexities amongst patient trajectories, it is very unlikely that the extent and the variety of the currently available data will cover the feasible range of physiologic states in any case. Further, it is well known that even survivors face a significant readmission risk and a reduced life expectancy [32, 47]. Therefore, not all survivors are the same, and we may have to consider the physiologic health or even a physiologic expected life time of the survivors when they are released.

The current intermediate reward choices are mostly adhoc, and typically it is not verified whether maximizing cumulative discounted rewards is a reasonable clinical goal. Further, there is enough evidence in RL where reasonable looking reward choices have caused undesirable or even dangerous behavior [3, 37]. In either case, the use of discount factors (which is necessary for mathematical guarantees) makes the interpretation of value functions opaque.

Thus, we propose a simple clinically motivated control objective for this problem : Maximizing the probability of surviving the ICU stay. We show how this objective could

then be interpreted as a Q learning based RL problem with patient state, and action specific discount terms. Thereby, allowing us to use any Deep Q learning based algorithm with a one line modification. The *Survival Q functions* also has a simple interpretation which can help in improving the trustworthiness of an RL agent, and provide *some* explainability of recommended actions.

Further, the same theoretical properties as Q learning hold under mild assumptions. We then experiment with this method using a large sepsis cohort and show qualitative differences between values and policies, compared with standard RL methods. We show that the scaled values are in particular, more consistent with clinical knowledge under our method.

In summary, in this article :

- We introduce a new, survival focused objective for critically-ill patients.
- We present theoretical results and then adapt this objective to a practical Deep RL algorithm.
- We experiment using a large sepsis cohort, and present how values are more consistent with clinical intuition under our scheme.

9.1 Related Work

As mentioned previously, there are a large volume of research which attempt to use RL or control for critical care applications [77, 63, 99, 133].

However, for the best of our knowledge there is limited prior work which explore alternate control objectives [1] or systematic criteria for defining RL rewards. [96] define a class of reward functions for which high-confidence policy improvement is possible. The authors, identity a space of reward functions that yield policies that are consistent in performance with the observed data. [89] learns a mortality risk score using semi supervised contrastive learning, and then use their risk score to define intermediate rewards as the decrease in risk between successive time steps.

[1] There have been risk sensitive RL methods, which optimize a different functional of the return rather than the expected value. However, these methods are also subject to a proper definition of rewards.

Arguably, the closest to our work is Q learning approaches for censored data such as [43]. However, their problem is fundamentally different to ours. They consider censored data, and define an objective which maximizes the survival time, taking the possible censoring into account. However, they focus on longer term problems and in contrast we focus on the shorter term, acute illnesses. We also have access to the end state of the patients, thus censoring isn't a major issue here.

Outside of medicine, [130] proposed a method, which aims to optimize the cumulative rewards in a constrained MDP, with a negative avoidance constraint. Their method uses a Negative Avoidance Function (NAF), which plays a role similar to a hazard function. However, apart from the higher level goal of prioritizing survival, the method proposed here is significantly different.

9.2 Background

We will start by briefly discussing the familiar RL framework and additive control objective. RL can be formalized by a Markov Decision Process (MDP) framework. This include state and action spaces $\mathcal{S}, \mathcal{A}$, a (typically unknown) Markov probability kernel $p(|s, a)$, which gives the dynamics of the next state, given the current state and the action and a reward process with a kernel $r(|s, a)$.

Given a discount factor γ, the return is defined as the cumulative discounted rewards : $\sum_{t=1}^{\infty} \gamma^t r_t$, which is a random variable. In RL, the agent's performance is measured in terms of the return, and most of the attention has been focused on the *expected* return.

Therefore, the *value* of a policy π ($V^\pi(s)$) at state s is defined as the expected future rewards starting from state s, and following the policy π. That is :

$$V^\pi(s) = \mathbb{E}_{p,\pi}[\Sigma_t \gamma^t r_t | s_0 = s, \pi], \quad \forall s \in \mathcal{S} \tag{37}$$

The Bellman equation for the value function can be written as:

$$V^\pi(s) = \mathbb{E}_{p,\pi}[r + \gamma V^\pi(s')], \tag{38}$$

If V^* is the optimal value function, V^* satisfies the following Bellman optimality equation:

$$V^*(s) = \sup_{\pi \in \Pi} \{\mathbb{E}_{p,\pi}[r + \gamma V^*(s')\} \tag{39}$$

Similarly, the *state action value function* or Q function can be defined as

$$Q^\pi(s,a) = \mathbb{E}_{p,\pi}[\Sigma_t \gamma^t r_t | s_0 = s, \pi, a_0 = a], \quad \forall s \in \mathcal{S}, a \in \mathcal{A} \tag{40}$$

The Q function can be interpreted as the expected return of starting at state s, taking the action a and then following the policy π.

The following can then be verified.

The Bellman equation for the Q function :

$$Q^\pi(s,a) = \mathbb{E}_p[r] + \gamma \mathbb{E}_{p,\pi}[Q^\pi(s',a')], \tag{41}$$

and the Bellman optimality equation for the Q function:

$$Q^*(s,a) = \mathbb{E}_p[r] + \gamma \mathbb{E}_p[\max_{a' \in A} Q^*(s',a')] \tag{42}$$

(where $Q^*(s,a)$ is the optimal Q function, and s' denotes the random next state)

Here, we have also implicitly assumed that the maximum exists for some $a \in \mathcal{A}$. If it doesn't, one can replace max with sup.

Indeed, it can be shown that under some regularity conditions all four Bellman operators are contractions in L^∞. So an iterative algorithm would converge to either the optimal value function or the policy induced value function.

9.3 Reinforcement Learning for Survival

An Idealized Objective for Critically Ill Patients

Now, we will present an idealized, clinically motivated control objective for critically ill patients. Our presentation will follow our intuition in developing the method. In particular, we will start by defining the objective without any consideration of its usefulness as a computational method, and then refine it so that it can be adapted to a RL algorithm, with convergence guarantees.

We will assume the knowledge of a true discrete time conditional hazard (or survival) process. That is : suppose a patient's death is a (Markov) stochastic process, based on the patient state, and a given action. Thus, for each patient state, at each time t there is a probability (discrete hazard) $h_t(s_t, a_t) = p(D_{t+1} = 1|s_t, a_t, D_t = 0)$ (where D_t=1 if the patient is dead at the end of the t th time step and 0 otherwise) of the patient dying within the next time step. We will further assume the hazard process is independent of the time t. Thus, we drop the subscript t from $h_t(s, a)$ from now on, assuming the hazard process is stationary, but of course state and action dependent.

Now, for a given policy π, it is straightforward to compute the expected probability of a patient surviving their ICU stay as :

$$\mathbb{E}_{p,\pi}[\prod_{t=0}^{H_s}(1 - h(s, a))|\pi] \tag{43}$$

Where, the expectation is taken with respect to the environment dynamics and the policy, the actions are $a_t \sim \pi(s_t)$ and H_s is a state dependent random time, representing the remaining time at the ICU.

Then, our control objective can be written as :

$$\text{Maximize, } \mathbb{E}_{p,\pi}[\prod_{t=0}^{H_s}(1 - h(s_t, a_t))] \text{ such that } \pi \in \Pi \tag{44}$$

Where Π, is the class of policies considered.

Notice that the functional represented by Equation 43 is multiplicative, but we will not be using it in the same form any further. However, we note that traditional stochastic control

literature have discussed multiplicative cost functionals [15]. That work discusses DP-like algorithms and guarantees of optimal policies which hold for our survival objective Equation 43 (under known dynamics and an uniform finite horizon).

However, we will take a different approach motivated by Q functions.

Let's define the *survival Q function* : $Q_S^\pi(s,a)$ to be the probability of a patient with state s will survive their ICU stay, given that the first action is a, and the policy π is continued afterwards.

Definition 1.

$$Q_S^\pi(s,a) := \mathbb{E}_{p,\pi}[\prod_{t=0}^{H_s}(1-h(s_t,a_t))|\pi, s_o = s, a_o = a] \tag{45}$$

Now analogous to Equation 42, we define the optimal survival Q functions as $Q_S^*(s,a)$:

Definition 2.

$$Q_S^*(s,a) := \sup_{\pi\in\pi} Q_S^\pi(s,a) \tag{46}$$

Now conditioning on the the event at $t=0$, the following two results follow immediately:

$$Q_S^\pi(s,a) = (1-h(s,a))\mathbb{E}_{p,\pi}[Q_S^\pi(s',a')] \tag{47}$$

$$Q_S^*(s,a) = (1-h(s,a))\mathbb{E}_{p}[\max_{a'\in\mathcal{A}} Q_S^*(s',a')] \tag{48}$$

With, for all $a \in \mathcal{A}$:

$Q_S^\pi(s,a), Q_S^*(s,a) = 1$, when s is a release state and,

$Q_S^\pi(s,a), Q_S^*(s,a) = 0$ when s is a death state.

Now, let R be an indicator variable such that $R(s) = 1$ if s is a release state, and 0 otherwise. We can interpret R as a known, deterministic binary function from $\mathcal{S} \to \{0,1\}$. [2] Then, Q^* satisfies the following relationship :

$$Q_S^*(s,a) = \mathbb{I}_{\{R(s)=1\}} + \mathbb{I}_{\{R(s)=0\}}(1-h(s,a))\mathbb{E}_{p}[\max_{a'\in A} Q_S^*(s',a')] \tag{49}$$

Implicit in Equation 49 is that for death states $h(s,a) = 1$, so we don't have to explicitly consider that case. Equation 49 allows us to develop simple RL and Deep RL algorithms for

[2] Alternatively, we can think of R as a known property or an annotation of the state

our problem. Before we describe the Deep RL algorithm we will present some theoretical results. For this, let's denote F to be the set of real valued functions from $\mathcal{S} \times \mathcal{A}$ to $\mathbb{R}$, and define the operators $T_\pi, T : F \to F$ as :

$$T_\pi(J)(s,a) = \mathbb{I}_{\{R(s)=1\}} + \mathbb{I}_{\{R(s)=0\}}(1 - h(s,a))\mathbb{E}_{p,\pi}[J(s',a')], \tag{50}$$

$$T(J)(s,a) = \mathbb{I}_{\{R(s)=1\}} + (1 - h(s,a))\mathbb{I}_{\{R(s)=0\}}\mathbb{E}_{p}[\max_{a' \in \mathcal{A}} J(s',a')], \tag{51}$$

Theorem 3. *Assume, the conditional hazard (at non-release states) is uniformly bounded below by a positive number. Then, the operators T_π and T are contractions in the Banach space B of bounded functions of F under the sup norm. Thus, they have unique fixed points.*

The proof of Theorem 3 follows with the exact same reasoning as results for analogous Bellman Q operators. However, we provide a proof in the Appendix A.

The contraction property of the optimal Survival Q function allows us to develop an experienced based, stochastic, *Survival Q learning* algorithm, akin to Q learning. This algorithm is guaranteed to converge under the same assumptions as Q learning. We relegate this theorem (Theorem 4) to Appendix A, due to space constraints.

As we noted earlier, the Survival Q function has a more straightforward interpretation than the regular Q functions (especially with intermediate rewards). That is : at each state s, and potential action a, $Q^*_S(s,a)$ represents the probability that the patient will survive their ICU stay, given that the action a is taken at this time step and actions are taken optimally afterwards. Therefore, the agent has some capacity to explain the reasoning of each decision it recommends. However, we note that the quality of the interpretation depends heavily on the quality of the function approximators, training data and the approximate hazard model. Still, we believe compared with the existing methods, this is one of the advantages of our method. We could also interpret our method as an uncertainty aware method, which penalizes *unlikely* survival by discounting the release by the likelihood of the survival, thus considering a form of aleatoric uncertainty. We will follow this insight and continue the discussion and possible modifications under Discussions.

Reinforcement Learning for Survival (RL4S)

Now, we can notice that Equation 49 can be compared with Equation 42, with zero intermediate *rewards*, deterministic terminal *rewards* and a state action specific *discount* factor. Since, we have the knowledge of the end outcome of terminal states, we can use this relationship exactly as DQN [87] type algorithms leverage Equation 42. More specifically, we aim to parametrize the optimal survival Q function (Q_S^*) using function approximation based on Equation 49. At terminal states the function is regressed into 1 or 0, and for every other state the left hand side is regressed to the the right hand side of Equation 2, with the same convergence tricks as DQN. [3]

This insight, allows us to leverage any value based Deep RL algorithm, with a *reward* [4] where a) a final reward of 1 is applied if and only if a patient is realised and b) 0 at all other time points. Whilst when interpreted as a reward, this choice is still sparse, using state, action specific survival probabilities instead of a uniform discounting term encodes information about the patient's condition.

9.4 Experiments

Now, we will conduct several experiments to investigate the performance of RL4S and to empirically compare the policies and values with other RL formulations. We will focus on the problem of administering vasopressors, and fluids for septic patients. This problem is well suited for our objective and is a popular choice for RL approaches [99, 63, 74, 61, 90]. However, we emphasise that our focus here is to investigate our method and thus our results are preliminary and doesn't include many necessary steps needed before it can used for practical clinical decision support. For example, we strongly believe that any application of computational methods for clinical decision support should include (especially epistemic) uncertainty quantification, however we don't explore such results here. In particular, we do

[3] Note, that this depends on a known hazard function, but there are several methods to learn an approximate hazard function, we will describe our choice in the experiments section.

[4] In our algorithm this is an indicator variable indicating if a patient has been released at the point or not. however the formulation fits into usual RL algorithms by interpreting this as a reward

not claim that the learned policies are superior to that of the clinicians or previous RL efforts.

9.4.1 Data Sources & Prepossessing

For all our analysis we used the MIMIC-III [56, 93] database and the same patient cohort which was used by Nanayakkara et al [90], including the representation learning described in that work. The cohort consisted of 18472 different patients out of which 1828 were non-survivors. All of these patients were adults (≥ 17), who satisfied the Sepsis 3 criteria [115]. The excluded patients included patients who died at the hospital, but after release from the ICU, and patients who had more than 25% missing values (vitals and scores) after creating hourly trajectories. This cohort resulted in 2596604 hourly transitions. The state space was 41 dimensional.

All the features were standardized for all work, and the missing values were imputed using a last value carried forward scheme, as long as the missingness was less than 25% after creating hourly trajectories. We used the 9 dimensional discrete action space used in [90].

9.4.2 RL4S

Since RL4S depends on a known hazard model, we first describe the approximate hazard model we used.

Hazard Model: We used a simple feed forward neural network (or multi-layer perceptron (MLP)) to estimate the conditional hazard. By definition, the conditional hazard is the probability of the event (in this case death) occurring within a time step, given that the event hasn't occurred previously. Therefore, using the Markov assumption, we frame this as a classification problem of predicting whether a patient would die within t and $t+1$, given the state s_t and the action a_t. To satisfy the iid assumption used in stochastic gradient descent, for each batch we first sampled the patients and then randomly sampled a patient state of that patient. To combat the heavy imbalanced nature of the problem (only 0.07% of states were death states), we sampled non-survivors more frequently, and for a non-survivor the patient state was taken to be the terminal state with 50% probability and a random state with 50% probability, Our architecture had two separate bases for the state and action and

then the two representations were combined and sent through another small MLP head.

We then adapted the existing Deep RL algorithms (We used the distributional C51 algorithm [9], but it is trivial to use any value based algorithm). The only change in implementation and design required is that instead of a uniform discount factor, we have to use a state and action specific survival probability, analogous to a discount factor (and defining a form of rewards as described previously).

In addition to RL4S, we also experimented with standard RL with terminal rewards of $+/-1$ depending of release or death and no intermediate rewards.

All the methods were trained using [9] for 7 epochs with the same hyper-parameters except the lower and upper limits of the approximating discrete distribution [5]. However, all methods displayed variation amongst recommended policies across weights saved after each epoch. Therefore for the value and policy results we present in the next section, we first averaged the value distributions of neural networks trained for 5,6 and 7 epochs.

9.5 Results

We will now discuss some results of the previously discussed experiments. We will start by investigating the Q values (Survival Q values for RL4S) of both methods.

First, we consider the averaged Q values (across actions and relevant states). Since the Q values are defined and scaled differently in each case, we used a max-min scaling scheme -so the scaled Q values are in between 0 and 1. We then stratified, these values by a) survivor, and non survivor states b) Last 24 hour states (before death or release) of each case. Figure 23 presents these results using box plots. Here the green boxes denotes the Survival Q values of RL4S, the yellow : Q values for standard RL with terminal-only rewards. Intuitively, we expect the Q values to capture the patient condition, and indicate the impending death or release at least when a patient is *close* to each.

We can notice that there is a significant separation between survivor and non-survivor Q values in RL4S. However, for RL with terminal-only rewards, even the median of the *last*

[5]These were taken to be 0 and 1 for Survival RL, -1.5 and 1.5 for terminal only RL

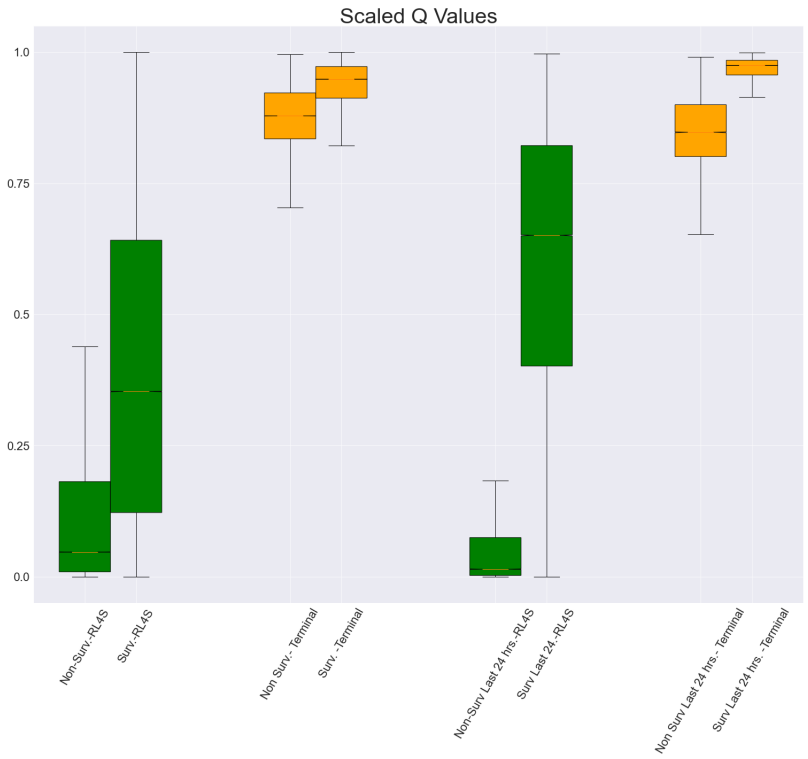


Fig. 23: Box plots of averaged Q values: For RL4S and standard RL and stratified by patient outcome

24 hr non -survivor scaled Q values is above 0.75. Considering the definition of the usual Q function (For terminal rewards: Ignoring the discounting, the Q value can be identified as a linear combination between expected release probability and expected death probability) this does not meet clinical intuition, as the models seem to be predicting survival even when the patients are close to death. In contrast RL4S in particular, seem to identify the higher mortality risk in advance.

We note that the main quantities of interest in RL algorithms, are not the values themselves but the difference between values of different actions. Therefore, it is possible for a method to overestimate Q values, and yet correctly identify the correct ordering of Q values (i.e. identify the optimal action order). However, explainability and trustworthiness are essential components of any automated medical decision making system. Value based algorithms attempt to learn optimal polices by estimating the values of states, and thus if

the values themselves are inconsistent with clinical knowledge and observed outcomes, such a system is unlikely to be trusted. Therefore the results of RL4S seem to be more promising in this aspect. It is also important to note that our patient cohort was heavily dominated by survivors. A more balanced cohort *could* result in more realistic Q values. Another possibility is to bias the sampling scheme as explained in [90], by sampling death and near death states with higher probability.

Next, we will discuss selected interesting properties of recommended actions. Note that for each state s, we select the action a, which maximizes the Q values. (i.e $a = \arg\max_{a' \in \mathcal{A}} Q(s, a')$). We will present the full global action distribution in the appendices.

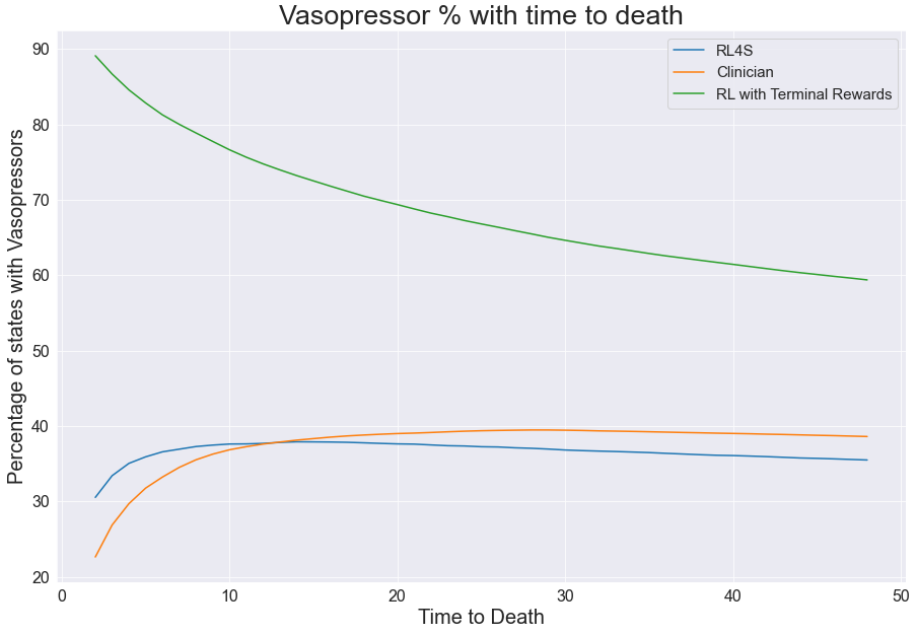


Fig. 24: Percentage of states with vasopressors: Recommended by RL and RL4S and administered by the clinicians

A striking observation is illustrated in figure 24. Here, we plot the percentages of states, with vasopressors recommended by each RL method, for non-survivors for different times to eventual death. Also, shown are the percentage of states for which the clinicians have used vasopressor therapy. The plots of RL4S and clinicians are remarkably similar, both even decrease as time to death decreases. However, for regular RL more vasopressors are recommended as patients approach death, which is consistent with the results presented in [90] for RL with intermediate rewards. They hypothesize that the decrease of states with

vasopressors given by clinicians may be due to decisions that were made by the patient's family to cease extraordinary measures. However, such information was not given to RL4S so it doesn't explain the behavior of RL4S. We plan to investigate the possible reasons in future work.

Unfortunately, evaluating policies in offline RL is an open problem with no satisfactory answers suited for critical care medicine [46]. Even, the current Off Policy Evaluation (OPE) methods are ill suited for intensive care medicine. Further, they are defined for a fixed reward choice making comparing policies under two different objectives even more complicated. Thus, we don't make any claims that policies under one schemes is necessarily better at this point.

9.6 Discussions & Conclusions

In this work, we introduced a control objective for RL applications in critical care medicine, which was motivated by the ambiguity of defining rewards. Indeed, the reward hypothesis is arguably the most fundamental component of RL and the only way to guide desired behavior of an agent. However, it is not immediate how rewards should be defined for most clinical decision making applications. Thus, we started from quantifying a reasonable clinical goal (i.e. maximizing the probability of survival) and developed a framework and an algorithm which can formalize this goal. We believe this objective is naturally suited to formalize the goal of reducing mortality. [6]

One limitation of our method, is that it depends on an approximate hazard model. For our experiments, we used a simple MLP in a supervised learning setting to estimate the conditional hazard. Also evaluation of survival models is more complicated than standard supervised learning methods. However, given that survival analysis is a well researched area, there are several alternatives, including methods where medical knowledge can be encoded. There are also ways to reduce the effect of the learned hazard method. For example, one could define a hybrid method which considers survival of a short term horizon and then use a

[6] Again, we emphasise that there are certainly other goals in critical care medicine, however we focus on problems where the primary goal is minimizing mortality risk. This certainly include a large class of problems.

look-ahead value learned using standard RL methods.

The similarity to Deep Q learning type algorithms, allows us to trivially implement a wide range of modifications and improvements to our method. For example, we can use most algorithms developed specifically for offline RL. (For example, [42]) Informally, these methods attempt to learn policies which are sufficiently close to the behavioral distribution. Additionally, we can use Equation 47 to define an Actor Critic method, instead of a pure value based method. Using distributional RL methods, we can naturally take environment uncertainty into account and modify Equation 44 by replacing the expectation operator by a risk sensitive measure (such as VAR or C-VAR) to define risk sensitive methods. In particular, methods designed for offline and risk sensitive problems such as [121], can be used.

Further, as we hinted earlier our objective has another interpretation which allows us to view it as an uncertainty aware method. To see this let's recall by Equation 2, our objective can be seen as a standard RL objective, with rewards given if and only if a patient is released, and at each time, instead of using a fixed discounting term, the probability of survival $1 - h(s, a)$ is used for discounting. Thus for each trajectory, the terminal reward is multiplied by the probability of surviving the ICU stay and thereby discounting unlikely releases more. This viewpoint allows us to investigate other avenues to incorporate Uncertainty Quantification, and possibly modify the objective.

Our initial experiments produced promising results. The Survival Q values seem to differentiate between survivor and non-survivor states and identify mortality risk in advance. However, as we have mentioned previously, comparing performance of different clinical RL methods using historical data is very challenging. Thus, further experiments and research have to be conducted before any stronger claims can be established. One possible way to evaluate the method would be to use a simulated environment of critically ill patients [7], and then compare the mortality rates under different methods, learned from a fixed set of trajectories. However, it is important to verify that any such environment will be sufficiently similar to the patient environment one is interested in, if not undesirable conclusions can follow. Thus, we defer these attempts to future work.

Finally, we note that stochastic control research has been historically dispersed amongst

[7] Or a different environment with similar goals of survival

various mathematics, computer science, operations research and artificial intelligence communities. However, recently there has been an effort to unify these efforts in to a single framework [95, 85]. We believe such an unified approach may result in methods specifically for healthcare and critical-care medicine.

9.7 Appendix A: Proof of Fixed Point Theorems

Proof. First notice that in either case the image of B is contained in B. i.e. $T, T_\pi : B \to B$.

We will first prove that $T_\pi, (50)$ is a contraction.

For ease of notation we will introduce the following notation $\beta(s,a) = (1 - h(s,a))$. Then note that by assumption, there exist $\gamma{<}1$ such that $\beta(s,a) < \gamma,\ \forall s, a$ with $R(s) = 0$.

Recall : $T_\pi : B \to B \quad T_\pi(s,a) = \mathbb{I}_{\{R(s)=1\}} + \mathbb{I}_{\{R(s)=0\}}(\beta(s,a))\mathbb{E}_{p,\pi}[J(s',a')]$

Thus, for $J, J' \in B$

$||T_\pi(J) - T_\pi(J')||_\infty$

$= \sup_{s\in\mathcal{S},a\in\mathcal{A}}|(T_\pi(J)(s,a) - T_\pi(J')(s,a)|$

$\leq \sup_{s\in\mathcal{S},a\in\mathcal{A}}|(\beta(s,a))\mathbb{E}_{p,\pi}((J)(s,a) - (J')(s,a))|$

$\leq \gamma \sup_{s\in\mathcal{S},a\in\mathcal{A}}|(J)(s,a) - (J')(s,a)|$

$= \gamma||J - J'||_\infty$

The second part regarding the unique fixed point follows directly from the Banach contraction theorem, and the completeness of B.

Now T is defined as :

$T(J)(s,a) = \mathbb{I}_{\{R(s)=1\}} + \mathbb{I}_{\{R(s)=0\}}(\beta(s,a))\mathbb{E}_p[\max_{a'\in\mathcal{A}} J(s',a')]$

First notice that for any two functions $f_1, f_2 : \mathcal{X} \to \mathbb{R}$

$$|\max_{x\in\mathcal{X}} f_1(x) - \max_{x\in\mathcal{X}} f_2(x)| \leq \max_{x\in\mathcal{X}}|f_1(x) - f_2(x)|$$

Then, for $J, J' \in B$ and $s \in \mathcal{S}, a \in \mathcal{A}$

$|T(J)(s,a) - T(J')(s,a)|$

$|(\beta(s,a)\mathbb{E}_p[(\max_{a\in\mathcal{A}}(J)(s,a)] - \mathbb{E}_p[(\max_{a\in\mathcal{A}}(J')(s,a))]|$

$= |\beta(s,a)\mathbb{E}_p[\max_{a\in\mathcal{A}}(J)(s,a) - \max_{a\in\mathcal{A}}(J')(s,a)]|$

$\leq (\beta(s,a))|\max_{a\in\mathcal{A}}(J)(s,a) - \max_{a\in\mathcal{A}}(J')(s,a))|$

$\leq (\beta(s,a))\max_{a\in\mathcal{A}}|(J)(s,a) - (J')(s,a))|$

$\leq ((\beta(s,a))\sup_{s\in\mathcal{S},a\in\mathcal{A}}|(J)(s,a) - (J)(s,a)|$

$< \gamma||J - J'||_\infty$

Now taking the supremum over $s \in \mathcal{S}, a \in \mathcal{A}$, we get that, $||T(J)-T(J')||_\infty \leq \gamma||J-J'||_\infty$

Again, the fixed point property follows.

□

9.8 Appendix B: Stochastic Approximation Theorem

Theorem 4. *If $(s_k, s'_k, a_k, h_k(s_k,a_k), R_k)$ $k \in \mathbb{N}$ is a set of experience tuples, generated from the underlying patient distribution. Where R is an indicator variable such that $R(s) = 1$ if the patient is released at this state and 0 otherwise.*

Suppose α_k, $k \in \mathbb{N}$ is a sequence of positive real numbers satisfying the Robbins Monro conditions [106], (for state, action pairs s_k, a_k) :

$\Sigma_{k=0}^{\infty}\mathbb{I}_{\{s=s_k,a=a_k\}}\alpha_k = \infty$ *and* $\Sigma_{k=0}^{\infty}\mathbb{I}_{\{s=s_k,a=a_k\}}\alpha_k^2 < \infty$. *with probability 1*

for all $s \in \mathcal{S}, a \in \mathcal{A}$.

Then, the algorithm defined by $Q_S^0(s,a) = 0$ and:

$Q^{k+1}(s,a) = (1-\alpha_k)Q_S^k(s,a) + (\alpha_k)\mathbb{I}_{\{s=s_k,a=a_k\}}[\mathbb{I}_{\{R(s)=1\}} + \mathbb{I}_{\{R(s)=0\}}\beta(s,a)\max_{a'\in\mathcal{A}}Q_S^k(s',a')]$

Converges to $Q_S^*(s,a)$ with probability 1.

The proof of the above theorem is also analogous to the corresponding convergence results of temporal difference methods and Q learning. However, a full proof, with the relevant background would be too lengthy for this text. We refer to [18, 11] for a general stochastic approximation results, and convergence proofs of Q Learning method [127].

9.9 Appendix C: Implementation Details

We used the standard C51 training algorithm as in [9]. Q network was a multi-layer neural network. We use a target network for all methods include RL4S, and update the target networks using polyak target updating with $\tau = 0.005$. (i.e. after every iteration/training step we set the target network weights to a linear combination of it's own weights, weighted by (1-τ) an the Q network weights, weighted by τ). This kind of target network is common amongst all deep Q learning, algorithms. We used the following hyper-parameters and optimization choices for the c-51 algorithm. As we mentioned previously, the maximum and minimum values of the approximating distribution and the discount factor for RL4S, were the only hyper-parameters which were not shared by all the methods.

Table 15: RL algorithm hyper-parameters

Hyper-Parameter	Value
Support size	51
γ	0.999
Batch size	124
Number of iterations	51932
Optimizer	Adam
Learning rate	3×10^{-4}
τ	0.005

As mentioned previously, the hazard model was treated as a standard classification problem. All the optimizations were conducted using Adam [62].

For both the hazard model and RL the state consisted of :

- **Demographics**: Age, Gender, Weight.
- **Vitals**: Heart Rate, Systolic Blood Pressure, Diastolic Blood Pressure, Mean Arterial Blood Pressure, Temperature, SpO2, Respiratory Rate.
- **Scores:** 24 hour based scores of, SOFA, Liver, Renal, CNS, Cardiovascular
- **Labs:** Anion Gap, Bicarbonate, Creatinine, Chloride, Glucose, Hematocrit, Hemoglobin, Platelet, Potassium, Sodium, BUN, WBC.
- **Latent States**: (see [90]) Cardiovascular states and 10 dimensional lab history representation.

9.10 Appendix D: RL4S: Recommended Actions

Table 16: Percentages of actions (Act.) recommended by RL4S and clinicians

Act.	RL4S	Clinician
Flu 0 Vaso 0	59.89	27.78
Flu 1 Vaso 0	3.79	23.70
Flu 2 Vaso 0	17.97	31.78
Flu 0 Vaso 1	3.29	1.29
Flu 1 Vaso 1	2.98	3.28
Flu 2 Vaso 1	10.57	3.98
Flu 0 Vaso 2	0.55	1.26
Flu 1 Vaso 2	0.91	2.51
Flu 2 Vaso 2	0.01	4.40

10.0 Conclusions

We have discussed opportunities and challenges in developing a computational toolbox to assist and direct clinical decision making for sepsis. The motivation and the potential benefits are clear and were mentioned in detail multiple times in the prequel. Thus, we will conclude this thesis by summarizing the work we presented and then discussing some further high level challenges and directions for future work: from both computational and medical perspectives.

We believe the work presented here, take considerable steps towards improving the current state of quantitative solutions to clinical sepsis decision making. We started from the control and RL framework itself, and then focused on a) framing the problem b) problems in defining the key components: states, objectives and c) associated uncertainties. We tried to address each, by using inspiration from a range of related but dispersed research fields. Of course, the problem of learning optimal treatment from data, is itself an inherently interdisciplinary problem. However, in our approaches we went even further by a) taking an unified view of control methods b) using first principle based mathematical modeling to encode domain knowledge and improve patient representation c) using uncertainty quantification to quantify parametric uncertainty d) taking a survival focused approach to the problem. We hope this work will be of interest to a wide range of research communities and will serve as an inspiration for more researchers from different backgrounds to work on this problem.

We also believe that we introduced new themes, perspectives, and concepts in our work, which we hope will have an impact beyond the methods presented here. For example, in Article 1, we introduced a neural network architecture that integrates a physiological model with a deep neural network, unifying two main modeling paradigms : first principle based mechanistic modeling and data driven machine learning methods. As we mentioned, there are numerous potential benefits of such a method. Further, we used uncertainty quantification and proposed a simple framework for decision making with humans in the loop. We note that we expect to build upon both of these ideas in future work. Uncertainty quantification is in particular essential for any practical clinical decision making system.

One of the main problems we focused on, was the ambiguity of the control objective and the lack of a clear notion of rewards. In Article 2, we used a semi-supervised learning method to learn mortality risk score: which then leads to a simple reward formulation. In the last article, we presented an alternate stochastic control objective for critically ill patients. At this point, we believe that both approaches provide clinically meaningful and interpretable objectives. However, they both share a common limitation: they depend on a learned hazard or a risk model.

We note that this work and *all* other related work is only a start. As we have mentioned previously there are a large number of obstacles that have to be addressed before the full potential of computational approaches can be realized. Despite, this we believe that the RL methods are close to being used in real time to *support* clinicians at the ICU. We make no claims of outperforming human clinicians in any foreseeable future. However, there is little doubt that using RL based support schemes can provide a number of strong benefits. If nothing else, a sufficiently large and representative dataset encodes experiences of a large number human clinicians. Of course, human clinicians' knowledge is certainly not limited to the experiences at the ICU. Therefore, it is up to the computational research communities to develop and validate methods to extract the most insight out of this data. Therefore, we believe that a highly useful area of work is to explore ways of encoding medical knowledge to RL based systems. One way of doing this is to use mathematical physiological models as we have done in this work, as these models embody decades of medical knowledge. From a machine learning perspective, concepts from continual learning, graph reasoning methods, self-supervised learning, multi-task and meta learning could be fruitful avenues to explore. Further, care to should be made to make the methods as interpretable, transparent and uncertainty aware as possible.

One significant challenge that we haven't proposed a solution is the problem of evaluation. Whilst the prospect of evaluating a learned policy exclusively using past data is appealing, for a problem as challenging as treating sepsis this is unlikely to suffice. Therefore we believe some form of clinical trails would be necessary. However, it is up to the medical community to decide when what and how this is done [105].

Whilst, the application of mathematical methods and computational tools to medicine

is hardly new, the modern AI based applications are still at an early phase. Over the past few years there has been an explosion of research that develop new methods or leverage existing methods for different clinical applications. Further, even when they focus on the same application (treating sepsis for example), there are non-trivial differences in the set up, methods, assumptions and the data (In RL for example: discrete vs continuous action spaces, state definitions, patient cohorts). Navigating such a large volume of work and potentially extracting the best out of them all in a consistent framework, is undoubtedly challenging. but is necessary.

This thesis was heavily quantitatively focused: both in questions and solutions. However, we re-emphasize that there are numerous ethical and social questions which have to be answered. It is extremely important that these problems are sufficiently addressed, and that a worldwide view is taken when these issues are considered. It is promising that there has been considerable interest in these aspects of AI, and we hope progress will be made in the near future to fully realize the potential of computational medicine, to the benefit of the whole world.

We now conclude this thesis, with an optimistic note. Improving the state of healthcare and medicine is one of the most virtuous goals of human and artificial intelligence. Computational and mathematical tools and current advances of technology make contributions to medicine more accessible to different academic communities. Humans in the loop medical AI has the potential to be an interface that could positively impact billions of lives and reshape medicine. We hope the work presented here, takes some small steps towards that ultimate goal in the context of clinical sepsis.

Appendix A Towards a Simulated Environment Using a Deep Probabilistic Mixture of Gaussians and a Survival Model

We have mentioned frequently the benefit of a simulated environment for septic patients with interventions. In this chapter, we will briefly our experiments on learning a deep generative model of septic dynamics. Although this model was successful in terms of quantitative evaluation (log likelihood and generating reasonable trajectories), some action induced results did not meet clinician intuition (for example the relationship between vasopressors and blood pressure). Therefore we want to verify and improve the model further before using it as a simulator.

We use the processed data described in Article 1. Then the model assumptions are simple to state :

We assume that given the state s_t (This state can potentially include a history representation), and action a_t, the next observation o_t [1]is distribution according to a mixture of two Gaussian densities. We parameterize the mixture probabilities and the Gaussian parameters by a neural network. Therefore it is then straightforward to leverage stochastic gradient based optimization methods and maximize the log likelihood of the data.

Formally:

$o_{t+1} \sim p(O|s_t, a_t)$ where $p(O|s_t, a_t) = \pi_\theta^1 \mathcal{N}(\mu_\theta^1, \Sigma_\theta^1) + \pi_\theta^2 \mathcal{N}(\mu_\theta^2, \Sigma_\theta^2)$

There is sufficient freedom to choose a neural network architecture. The simplest method would be to use a Markov assumption, and define the state to be the same as observables.

To simulate a septic patient we also need a mortality process. For this we use the survival model which we described in Article 3. Briefly, use a Markov assumption and parameterize the conditional hazard (the probability of dying within the next time step) by a MLP. Then the problem is reduced to a classification problem. The trained model predicts a patient would die within the next hour, given the current state and the action taken.

Given the probabilistic model for the environment and the survival model, it's easy to

[1]We make the state, and observations distinct so we could use the same notation for different ways of framing the problem. For example, when we use a recurrent neural network (RNN), the state includes the hidden representation produced by the RNN but of course this is not predicted by the model.

simulate patients. However, the challenge is of course to verify the accuracy of the model beyond the log-likelihood. Therefore, given that the work is incomplete we keep our discussion brief, however we will share our implementation (see next appendix), so the model may be improved independently.

Appendix B Code Repository

Research level code for all the experiments described are publicly available in the following repositories.

- `https://github.com/thxsxth/POMDP_RLSepsis`
- `https://github.com/thxsxth/normed_constrastive_metric`
- `https://github.com/thxsxth/survRL`
- `https://github.com/thxsxth/DMG`

Bibliography

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Abbas Khosravi, U Rajendra Acharya, Vladimir Makarenkov, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *arXiv preprint arXiv:2011.06225*, 2020.

[2] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, 2020.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

[4] Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, 297:103500, 2021.

[5] Nicolas Baradel, Bruno Bouchard, and Ngoc Minh Dang. Optimal control under uncertainty and bayesian parameters adjustments. *SIAM Journal on Control and Optimization*, 56(2):1038–1057, 2018.

[6] Gabriel Barth-Maron, Matthew W Hoffman, David Budden, Will Dabney, Dan Horgan, Dhruva Tb, Alistair Muldal, Nicolas Heess, and Timothy Lillicrap. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

[7] Edmon Begoli, Tanmoy Bhattacharya, and Dimitri Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence*, 1(1):20–23, 2019.

[8] Edmon Begoli, Tanmoy Bhattacharya, and Dimitri F. Kusnezov. The need for uncertainty quantification in machine-assisted medical decision making. *Nature Machine Intelligence (Online)*, 1(1), 1 2019.

[9] Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. *arXiv preprint arXiv:1707.06887*, 2017.

[10] Marc G. Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of*

the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, pages 449–458, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[11] Marc G. Bellemare, Will Dabney, and Mark Rowland. *Distributional Reinforcement Learning*. MIT Press, 2022. `http://www.distributional-rl.org`.

[12] Richard Bellman and Robert E Kalaba. *Dynamic programming and modern control theory*, volume 81. Citeseer, 1965.

[13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[14] Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. The modern mathematics of deep learning. *arXiv preprint arXiv:2105.04026*, 2021.

[15] Dimitri P Bertsekas and Steven E Shreve. *Stochastic optimal control: the discrete-time case*, volume 5. Athena Scientific, 1996.

[16] Ramin Bighamian, Andrew T Reisner, and Jin-Oh Hahn. An analytic tool for prediction of hemodynamic responses to vasopressors. *IEEE Transactions on Biomedical Engineering*, 61(1):109–118, 2013.

[17] Ramin Bighamian, Sadaf Soleymani, Andrew T Reisner, Istvan Seri, and Jin-Oh Hahn. Prediction of hemodynamic response to epinephrine via model-based system identification. *IEEE journal of biomedical and health informatics*, 20(1):416–423, 2014.

[18] Vivek S Borkar. *Stochastic approximation: a dynamical systems viewpoint*, volume 48. Springer, 2009.

[19] Michael M Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.

[20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[21] Lucian Busoniu, Robert Babuska, Bart De Schutter, and Damien Ernst. *Reinforcement learning and dynamic programming using function approximators*. CRC press, 2017.

[22] João Caldeira and Brian Nord. Deeply uncertain: comparing methods of uncertainty quantification in deep learning algorithms. *Machine Learning: Science and Technology*, 2(1):015002, 2020.

[23] Junyi Chai, Hao Zeng, Anming Li, and Eric WT Ngai. Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Machine Learning with Applications*, 6:100134, 2021.

[24] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE, 2016.

[25] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[26] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[27] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[28] Yuyang Chen, Kaiming Bi, Chih-Hang John Wu, and David Ben-Arieh. A new evidence-based optimal control in healthcare delivery: a better clinical treatment management for septic patients. *Computers & Industrial Engineering*, 137:106010, 2019.

[29] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[30] Jongeun Choi and Dejan Milutinović. Tips on stochastic optimal feedback control and bayesian spatiotemporal models: Applications to robotics. *Journal of Dynamic Systems, Measurement, and Control*, 137(3):030801, 2015.

[31] Gilles Clermont, John Bartels, Rukmini Kumar, Greg Constantine, Yoram Vodovotz, and Carson Chow. In silico design of clinical trials: A method coming of age. *Critical Care Medicine*, 32, 2004.

[32] Brian H Cuthbertson, Andrew Elders, Sally Hall, Jane Taylor, Graeme MacLennan, Fiona Mackirdy, and Simon J Mackenzie. Mortality and quality of life in the five years after severe sepsis. *Critical Care*, 17(2):1–8, 2013.

[33] Will Dabney, Mark Rowland, Marc Bellemare, and Rémi Munos. Distributional reinforcement learning with quantile regression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[34] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[36] Ivor S Douglas, Philip M Alapat, Keith A Corl, Matthew C Exline, Lui G Forni, Andre L Holder, David A Kaufman, Akram Khan, Mitchell M Levy, Gregory S Martin, et al. Fluid response evaluation in sepsis hypotension and shock: A randomized clinical trial. *Chest*, 2020.

[37] Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese*, 198(27):6435–6467, 2021.

[38] Flavio Lopes Ferreira, Daliana Peres Bota, Annette Bross, Christian Mélot, and Jean-Louis Vincent. Serial evaluation of the sofa score to predict outcome in critically ill patients. *Jama*, 286(14):1754–1758, 2001.

[39] Paul Festor, Giulia Luise, Matthieu Komorowski, and A Aldo Faisal. Enabling risk-aware reinforcement learning for medical interventions through uncertainty decomposition. *arXiv preprint arXiv:2109.07827*, 2021.

[40] Pierre Foulon and Daniel De Backer. The hemodynamic effects of norepinephrine: far more than an increase in blood pressure! *Annals of translational medicine*, 6(Suppl 1), 2018.

[41] Florian Fuchs, Yunlong Song, Elia Kaufmann, Davide Scaramuzza, and Peter Duerr. Super-human performance in gran turismo sport using deep reinforcement learning. *arXiv preprint arXiv:2008.07971*, 2020.

[42] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062, 2019.

[43] Yair Goldberg and Michael R Kosorok. Q-learning with censored data. *Annals of statistics*, 40(1):529, 2012.

[44] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[45] 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, 2019.

[46] Omer Gottesman, Fredrik Johansson, Joshua Meier, Jack Dent, Donghun Lee, Srivatsan Srinivasan, Linying Zhang, Yi Ding, David Wihl, Xuefeng Peng, et al. Evaluating reinforcement learning algorithms in observational health settings. *arXiv preprint arXiv:1805.12298*, 2018.

[47] Raquel Bragante Gritte, Talita Souza-Siqueira, Rui Curi, Marcel Cerqueira Cesar Machado, and Francisco Garcia Soriano. Why septic patients remain sick after hospital discharge? *Frontiers in Immunology*, 11:3873, 2021.

[48] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[49] Alexandre Heuillet, Fabien Couthouis, and Natalia Díaz-Rodríguez. Explainability in deep reinforcement learning. *Knowledge-Based Systems*, 214:106685, 2021.

[50] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International workshop on similarity-based pattern recognition*, pages 84–92. Springer, 2015.

[51] Wei Hu, Lechao Xiao, and Jeffrey Pennington. Provable benefit of orthogonal initialization in optimizing deep linear networks. *arXiv preprint arXiv:2001.05992*, 2020.

[52] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[53] Dominik Jarczak, Stefan Kluge, and Axel Nierhaus. Sepsis—pathophysiology and therapeutic concepts. *Frontiers in Medicine*, 8, 2021.

[54] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[55] Alistair EW Johnson, Jerome Aboab, Jesse D Raffa, Tom J Pollard, Rodrigo O Deliberato, Leo A Celi, and David J Stone. A comparative analysis of sepsis identification methods in an electronic database. *Critical care medicine*, 46(4):494–499, 2018.

[56] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3:160035, 2016.

[57] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[58] Gregory Kahn, Adam Villaflor, Vitchyr Pong, Pieter Abbeel, and Sergey Levine. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv preprint arXiv:1702.01182*, 2017.

[59] Kenneth L Kehl, Haitham Elmarakeby, Mizuki Nishino, Eliezer M Van Allen, Eva M Lepisto, Michael J Hassett, Bruce E Johnson, and Deborah Schrag. Assessment of deep natural language processing in ascertaining oncologic outcomes from radiology reports. *JAMA oncology*, 5(10):1421–1429, 2019.

[60] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

[61] Taylor W Killian, Haoran Zhang, Jayakumar Subramanian, Mehdi Fatemi, and Marzyeh Ghassemi. An empirical study of representation learning for reinforcement learning in healthcare. In *Machine Learning for Health*, pages 139–160. PMLR, 2020.

[62] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[63] 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.

[64] Benjamin Kompa, Jasper Snoek, and Andrew L Beam. Second opinion needed: communicating uncertainty in medical machine learning. *NPJ Digital Medicine*, 4(1):1–6, 2021.

[65] Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013.

[66] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[67] Isotta Landi, Benjamin S Glicksberg, Hao-Chih Lee, Sarah Cherng, Giulia Landi, Matteo Danieletto, Joel T Dudley, Cesare Furlanello, and Riccardo Miotto. Deep representation learning of electronic health records to unlock patient stratification at scale. *NPJ digital medicine*, 3(1):1–11, 2020.

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

[69] Adam Laud and Gerald DeJong. The influence of reward on the speed of reinforcement learning: An analysis of shaping. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 440–447, 2003.

[70] Alexandra Lazăr, Anca Meda Georgescu, Alexander Vitin, and Leonard Azamfirei. Precision medicine and its role in the treatment of sepsis: a personalised view. *The Journal of Critical Care Medicine*, 5(3):90–96, 2019.

[71] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.

[72] Yann LeCun and Ishan Misra. Self-supervised learning: The dark matter of intelligence, 2021.

[73] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[74] 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.

[75] Michael Lederman Littman. *Algorithms for sequential decision-making*. Brown University, 1996.

[76] Ran Liu, Joseph L Greenstein, James C Fackler, Jules Bergmann, Melania M Bembea, and Raimond L Winslow. Offline reinforcement learning with uncertainty for treatment strategies in sepsis. *arXiv preprint arXiv:2107.04491*, 2021.

[77] Siqi Liu, Kay Choong See, Kee Yuan Ngiam, Leo Anthony Celi, Xingzhi Sun, and Mengling Feng. Reinforcement learning for clinical decision support in critical care: comprehensive review. *Journal of medical Internet research*, 22(7):e18477, 2020.

[78] Vincent Liu, Gabriel J Escobar, John D Greene, Jay Soule, Alan Whippy, Derek C Angus, and Theodore J Iwashyna. Hospital deaths in patients with sepsis from 2 independent cohorts. *Jama*, 312(1):90–92, 2014.

[79] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017.

[80] Björn Lötjens, Michael Everett, and Jonathan P How. Safe reinforcement learning with model uncertainty estimates. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8662–8668. IEEE, 2019.

[81] Xinrui Lyu, Matthias Hueser, Stephanie L Hyland, George Zerveas, and Gunnar Raetsch. Improving clinical predictions through unsupervised time series representation learning. *arXiv preprint arXiv:1812.00490*, 2018.

[82] P Marik and Rinaldo Bellomo. A rational approach to fluid therapy in sepsis. *BJA: British Journal of Anaesthesia*, 116(3):339–349, 2016.

[83] Paul E Marik, Liam Byrne, and Frank Van Haren. Fluid resuscitation in sepsis: the great 30 ml per kg hoax. *Journal of Thoracic Disease*, 12(Suppl 1):S37, 2020.

[84] PE Marik. The demise of early goal-directed therapy for severe sepsis and septic shock. *Acta Anaesthesiologica Scandinavica*, 59(5):561–567, 2015.

[85] Sean Meyn. *Control Systems and Reinforcement Learning*. Cambridge University Press, 2022.

[86] Riccardo Miotto, Li Li, Brian A Kidd, and Joel T Dudley. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 6(1):1–10, 2016.

[87] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[88] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

[89] Thesath Nanayakkara, Gilles Clermont, Christopher James Langmead, and David Swigon. Deep normed embeddings for patient representation. *arXiv preprint arXiv:2204.05477*, 2022.

[90] 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, 2022.

[91] Carly J. Paoli, Mark A. Reynolds, Meenal Sinha, Matthew Gitlin, and Elliott Crouser. Epidemiology and costs of sepsis in the united states—an analysis based on timing of diagnosis and severity level*. *Critical Care Medicine*, 46(12):1889–1897, 2018.

[92] 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.

[93] Alistair EW Pollard, Tom J abd Johnson. The mimic-iii clinical database. `http://dx.doi.org/10.13026/C2XW26`, 2016.

[94] Paul Porter, Udantha Abeyratne, Vinayak Swarnkar, Jamie Tan, Ti-wan Ng, Joanna M Brisbane, Deirdre Speldewinde, Jennifer Choveaux, Roneel Sharan, Keegan Kosasih, et al. A prospective multicentre study testing the diagnostic accuracy of an automated cough sound centred analytic system for the identification of common respiratory disorders in children. *Respiratory research*, 20(1):1–10, 2019.

[95] Warren B Powell. *Reinforcement Learning and Stochastic Optimization: A unified framework for sequential decisions*. John Wiley & Sons, 2022.

[96] Niranjani Prasad, Barbara Engelhardt, and Finale Doshi-Velez. Defining admissible rewards for high-confidence policy evaluation in batch reinforcement learning. In *Proceedings of the ACM Conference on Health, Inference, and Learning*, pages 1–9, 2020.

[97] Doina Precup. Eligibility traces for off-policy policy evaluation. *Computer Science Department Faculty Publication Series*, page 80, 2000.

[98] Erika Puiutta and Eric Veith. Explainable reinforcement learning: A survey. In *International cross-domain conference for machine learning and knowledge extraction*, pages 77–95. Springer, 2020.

[99] Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017.

[100] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[101] Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference. *Proceedings of Robotics: Science and Systems VIII*, 2012.

[102] Chanu Rhee, Raymund Dantes, Lauren Epstein, David J Murphy, Christopher W Seymour, Theodore J Iwashyna, Sameer S Kadri, Derek C Angus, Robert L Danner, Anthony E Fiore, et al. Incidence and trends of sepsis in us hospitals using clinical vs claims data, 2009-2014. *Jama*, 318(13):1241–1249, 2017.

[103] Andrew Rhodes, Laura E Evans, Waleed Alhazzani, Mitchell M Levy, Massimo Antonelli, Ricard Ferrer, Anand Kumar, Jonathan E Sevransky, Charles L Sprung, Mark E Nunnally, et al. Surviving sepsis campaign: international guidelines for management of sepsis and septic shock: 2016. *Intensive care medicine*, 43(3):304–377, 2017.

[104] Elsa Riachi, Muhammad Mamdani, Michael Fralick, and Frank Rudzicz. Challenges for reinforcement learning in healthcare. *arXiv preprint arXiv:2103.05612*, 2021.

[105] Samantha Cruz Rivera, Xiaoxuan Liu, An-Wen Chan, Alastair K Denniston, and Melanie J Calvert. Guidelines for clinical trial protocols for interventions involving artificial intelligence: the spirit-ai extension. *bmj*, 370, 2020.

[106] Herbert Robbins and Sutton Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[107] Mark Rowland, Marc Bellemare, Will Dabney, Remi Munos, and Yee Whye Teh. An analysis of categorical distributional reinforcement learning. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 29–37, Playa Blanca, Lanzarote, Canary Islands, 09–11 Apr 2018. PMLR.

[108] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[109] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[110] Matthew W Semler and Todd W Rice. Sepsis resuscitation: fluid choice and dose. *Clinics in chest medicine*, 37(2):241–250, 2016.

[111] Robin Senge, Stefan Bösner, Krzysztof Dembczyński, Jörg Haasenritter, Oliver Hirsch, Norbert Donner-Banzhoff, and Eyke Hüllermeier. Reliable classification: Learning classifiers that distinguish aleatoric and epistemic uncertainty. *Information Sciences*, 255:16–29, 2014.

[112] Rui Shi, Olfa Hamzaoui, Nello De Vita, Xavier Monnet, and Jean-Louis Teboul. Vasopressors in septic shock: which, when, and how much? *Annals of Translational Medicine*, 8(12), 2020.

[113] Yuqi Si, Jingcheng Du, Zhao Li, Xiaoqian Jiang, Timothy Miller, Fei Wang, W Jim Zheng, and Kirk Roberts. Deep representation learning of patient data from electronic health records (ehr): A systematic review. *Journal of Biomedical Informatics*, 115:103671, 2021.

[114] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[115] 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, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810, 2016.

[116] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[117] Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148. PMLR, 2016.

[118] Philip S Thomas, Georgios Theocharous, and Mohammad Ghavamzadeh. High-confidence off-policy evaluation. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[119] Hans-Christian Thorsen-Meyer, Annelaura B Nielsen, Anna P Nielsen, Benjamin Skov Kaas-Hansen, Palle Toft, Jens Schierbeck, Thomas Strøm, Piotr J Chmura, Marc Heimann, Lars Dybdahl, et al. Dynamic and explainable machine learning prediction of mortality in patients in the intensive care unit: a retrospective study of high-frequency data in electronic patient records. *The Lancet Digital Health*, 2(4):e179–e191, 2020.

[120] Nenad Tomašev, Xavier Glorot, Jack W Rae, Michal Zielinski, Harry Askham, Andre Saraiva, Anne Mottram, Clemens Meyer, Suman Ravuri, Ivan Protsyuk, et al. A clinically applicable approach to continuous prediction of future acute kidney injury. *Nature*, 572(7767):116–119, 2019.

[121] Núria Armengol Urpí, Sebastian Curi, and Andreas Krause. Risk-averse offline reinforcement learning, 2021.

[122] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[123] Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pages 831–838, 1992.

[124] J-L Vincent, Rui Moreno, Jukka Takala, Sheila Willatts, Arnaldo De Mendonça, Hajo Bruining, CK Reinhart, PeterM Suter, and Lambertius G Thijs. The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure, 1996.

[125] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[126] Jason Waechter, Anand Kumar, Stephen E Lapinsky, John Marshall, Peter Dodek, Yaseen Arabi, Joseph E Parrillo, R Phillip Dellinger, Allan Garland, Cooperative Antimicrobial Therapy of Septic Shock Database Research Group, et al. Interaction between fluids and vasoactive agents on mortality in septic shock: a multicenter, observational study. *Critical care medicine*, 42(10):2158–2168, 2014.

[127] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

[128] Lindsay Wells and Tomasz Bednarz. Explainable ai and reinforcement learning—a systematic review of current approaches and trends. *Frontiers in artificial intelligence*, 4:48, 2021.

[129] Alex Xiao, Christian Fuegen, and Abdelrahman Mohamed. Contrastive semi-supervised learning for asr. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3870–3874. IEEE, 2021.

[130] Changkun Ye, Huimin Ma, Xiaoqin Zhang, Kai Zhang, and Shaodi You. Survival-oriented reinforcement learning model: An effcient and robust deep reinforcement learning algorithm for autonomous driving problem. In *International Conference on Image and Graphics*, pages 417–429. Springer, 2017.

[131] Hugo Yèche, Gideon Dresdner, Francesco Locatello, Matthias Hüser, and Gunnar Rätsch. Neighborhood contrastive learning applied to online patient monitoring. In *International Conference on Machine Learning*, pages 11964–11974. PMLR, 2021.

[132] JD Young. The heart and circulation in severe sepsis. *British journal of anaesthesia*, 93(1):114–120, 2004.

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

[134] X. Zhang. Structural risk minimization. In *Encyclopedia of Machine Learning*, 2010.