Medical Dead-ends and Learning to Identify High-risk States and Treatments

Anonymous Author(s)
Affiliation
Address
email

Abstract

Machine learning has successfully framed many sequential decision making problems as either supervised prediction, or optimal decision-making policy identification via reinforcement learning. In data-constrained offline settings, both approaches may fail as they assume fully optimal behavior or rely on exploring alternatives that may not exist. We introduce an inherently different approach that identifies "dead-ends" of a state space. We focus on patient condition in the intensive care unit, where a "medical dead-end" indicates that a patient will expire, regardless of all potential future treatment sequences. We postulate "treatment security" as avoiding treatments with probability proportional to their chance of leading to dead-ends, present a formal proof, and frame discovery as an RL problem. We then train three independent deep neural models for automated state construction, dead-end discovery and confirmation. Our empirical results discover that dead-ends exist in real clinical data among septic patients, and further reveal gaps between secure treatments and those administered.

1 Introduction

Off-policy Reinforcement Learning (RL) was designed as the way to isolate behavioural policies, which generate experience, from the target policy, which aims for optimality. It also enables learning multiple target policies with different goals from the same data-stream or from previously recorded experience [1]. This algorithmic approach is of particular importance in safety-critical domains such as robotics [2], education [3] or healthcare [4] where data collection should be regulated as it is expensive or carries significant risk. Despite significant advances made possible by off-policy RL combined with deep neural networks [5–7], the performance of these algorithms degrade drastically in fully offline settings, without additional interactions with the environment [8, 9]. These challenges are deeply amplified when the dataset is limited and exploratory new data cannot be collected for ethical or safety purposes. This is because robust identification of an optimal policy requires exhaustive trial and error of various courses of actions [10, 11]. In such fully offline cases, naively learned policies may significantly overfit to data-collection artifacts [12–14]. Estimation errors due to limited data may further lead to mistimed or inappropriate decisions with adverse safety consequences [15].

Even if optimality is not attainable in such constrained cases, negative outcomes in data can be used to identify behaviors to avoid, thereby guard against overoptimistic decisions in safety-critical domains that may be significantly biased due to reduced data availability. In one such domain, healthcare, RL has been used to identify optimal treatment policies based on observed outcomes of past treatments [16]. These policies correspond to advising *what treatments to administer*, given a patient's condition. Unfortunately, exploration of potential courses of treatment is not possible in most clinical settings due to legal and ethical implications; hence, RL estimates of optimal policies are largely unreliable in healthcare [17].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

In this paper, we develop a novel RL-based method, Dead-end Discovery (DeD), to identify *treatments to avoid* as opposed to what treatment to select. DeD replaces the untenable problem of finding an optimal treatment policy by limiting the scope of a given policy based on retrospective analysis of observed outcomes. Our paradigm shift avoids pitfalls that may arise by constraining policies to remain close to possibly suboptimal recorded behavior as is typical in current state of the art offline RL approaches [9, 18–20]. When the data significantly lacks exploration, these methods fail to attain a reliable policy. We instead use such data to limit the scope of the policy, a more tractable approach when little data is provided. Our goal is to avoid dead-ends (formally defined in Section 3.2) or regions in the state space from which negative outcomes are inevitable. DeD identifies dead-ends via two complementary Markov Decision Processes (MDPs) using Deep Q-Networks (DQN) [5] to independently estimate likelihood of a negative outcome occurring (D-network) and the reachability of a positive outcome (R-network). Altogether DeD formally connects the notion of value functions to the dead-end problem, learned directly from offline data.

We validate DeD in a carefully constructed toy domain, and then evaluate real health records of septic patients in an intensive care unit (ICU) setting [21]. Sepsis treatment and onset is a common task in medical RL [22–25] because the condition is highly prevalent [26, 27], physiologically severe [28], costly [29] and poorly understood [30]. Notably, the treatment of sepsis itself may also contribute to a patient's deterioration [31, 32], thus making treatment avoidance a particularly well-suited objective. We find that DeD confirms the existence of dead-ends. We demonstrate that 11% of treatments administered to terminally ill patients are detrimental to their chances of survival, some occurring as early as 24 hours prior to death. The value function estimates underlying DeD are able to capture significant deterioration in patient health 4 to 8 hours ahead of observed clinical interventions, and that higher-risk treatments possibly account for this delay. Early identification of suboptimal treatment options is of great importance since sepsis treatment has shown multiple interventions within tight time frames (10 to 180 minutes) after suspected onset decreases sepsis mortality [33].

While motivated by healthcare, we propose the use of DeD in safety-critical applications of RL in most data-constrained settings. We construct and train DeD in a generic manner which can readily be used for other data-constrained sequential decision-making problems. In particular, we emphasize that DeD is well suited to analyze high-risk decisions in real-world domains.

2 Related Work

RL in Health: RL has been the subject of much focus in health [16], with particular emphasis on sepsis seeking to develop optimal treatment recommendation policies [22–25, 34–37]. However, with fixed retrospective medical data, an optimal policy that maximizes a patient's chance of recovery is both computationally and experimentally infeasible. To our knowledge, we are the first to target improved treatment recommendations by avoiding high-risk treatments in a fully offline manner.

Safety in RL: RL has a rich history in safety [38], with recent work attempting to limit high risk actions by constraining parametric uncertainty [39], through alignment between agent and human objectives [40, 41], by directly constraining the agent optimization process to avoid unsafe actions [42], or by improving over a baseline policy [43]. In these settings model performance is evaluated in online settings where more data can be acquired or models can be tested against new cases as well as known baselines. We focus on the more challenging offline setting with limited and non-exploratory data, reflecting the reality of healthcare settings.

Dead-ends: The concept of dead-ends and the corresponding security condition that we build from was proposed by Fatemi et al. [44] in the context of *exploration*. In their work an online RL agent needs to experience various courses of actions from each state, through which it learns optimal behavior. We adapt this approach and expand the theoretical results to offline RL settings such as those found within healthcare–where exploration is untenable–to determine which treatments increase the likelihood of entering a dead-end, based on the patient's current health state.

3 Methods

3.1 Preliminaries

Our pipeline isolates state construction from value estimation with RL. Therefore we consider episodic Markov Decision Processes (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the discrete

sets of states and treatments[1]; $T: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ is a function that defines the probability of transitioning from state s_t to s_{t+1} if treatment a_t is administered; $R: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [r_{min}, r_{max}]$ is a finite reward function and $\gamma \in [0,1]$ denotes a scalar discount factor.

A *policy* $\pi(s,a) = \mathbb{P}[A_t = a | S_t = s]$ defines how treatments are selected when given a state. A *trajectory* is comprised of sequences of (S_t, A_t, R_t, S_{t+1}) with S_0 being the initial state of the trajectory. Sequential application of the policy is used to construct trajectories. The reward collected over the course of a trajectory induces the *return* $G_t = \sum_{j=0}^{\infty} \gamma^j R_{t+j+1}$. We assume that all the returns are finite and bounded. A trajectory is considered *optimal* if its return is maximized. A state-treatment value function $Q^\pi(s,a) = \mathbb{E}^\pi[G_0 | S_0 = s, A_0 = a]$ is defined in conjunction with a policy π to evaluate the expected return of administering treatment a at state s and following π thereafter. The optimal state-treatment value function is defined as $Q^*(s,a) = \max_\pi Q^\pi(s,a)$, which is the maximum expected return of all trajectories starting from (s,a). We define state value and optimal state value as $V^\pi(s) = \mathbb{E}_{a \sim \pi} Q^\pi(s,a)$ and $V^*(s) = \max_a Q^*(s,a)$.

3.2 Special States

We define a terminal state as the final observation of any recorded trajectory. We focus on two types of terminal state that correspond to positive or negative outcomes. Our goal is to identify all *dead-end* states, from which negative outcomes are unavoidable, regardless of future treatments. In safety-critical domains, it is crucial to avoid such states *and* identify the probability with which any available treatment will lead to a dead-end. We also introduce the complementary concept of *rescue* states, from which a positive outcome is *reachable* with probability one. If an agent is in a rescue state, there exists at least one treatment at each time step afterwards which leads to either another rescue state or the eventual positive outcome. The fundamental contrast between dead-end and rescue states is that if the agent enters a rescue state, it *does not* mean the treatment process is done; it rather means that at each time step afterwards there exists at least one treatment to be found and administered until the positive outcome occurs. There might be trajectories starting from a rescue state which include non-rescue states. This is not the case for a dead-end state.

Formally, we augment $\mathcal{M}$ with a non-empty termination set $\mathcal{S}_T \subset \mathcal{S}$, which is the set of all terminal states. Mathematically, a terminal state is absorbing (self-transition w.p. 1) with zero reward afterwards. All terminal states are by definition zero-valued, but the transitions to them may be associated with a non-zero reward. We require that, from all states, there exists at least one trajectory with non-zero probability arriving at a terminal state. In an offline setting with limited and non-exploratory data, inducing an optimal policy *is not feasible* [11]; hence, we do not specify the reward function of $\mathcal{M}$. Finally, the sets of dead-end and rescue states are denoted by $\mathcal{S}_D$ and $\mathcal{S}_R$, respectively. We formally distinguish dead-end/rescue from the outcome, and assert that $\mathcal{S}_D, \mathcal{S}_R \not\subset \mathcal{S}_T$.

3.3 Treatment Security

When dealing with data-constrained offline scenarios a core distinction is necessary: Realization of an optimal treatment at a given state requires knowledge of all future outcomes for all possible treatments, which is not feasible. However, the data may contain enough information to estimate the possible outcome of a certain treatment at a similar state. If such an outcome is negative with high probability, then we should advise against the treatment, even if an optimal treatment still remains unknown. This distinction leads to a paradigm shift from finding the best possible treatment (which is untenable) to mindful avoidance of dangerous ones. This shift further motivates a different design space to make use of the limited, yet available data.

We adapt the security condition from Fatemi et al. [44] and formally frame the treatment avoidance problem with a more generalized *treatment security condition*. We note that the chance of a negative outcome is best described by the probability of falling in a dead-end or immediate negative termination. The security condition therefore constrains the scope of a given behavioral policy π if *any* knowledge exists about dead-ends or negative termination. Formally, if at state s, treatment a leads to a dead-end with probability $P_D(s,a)$ or immediate negative termination with probability $F_D(s,a)$ with a level

[1] Our results can easily be extended to continuous state-spaces by properly replacing summations with integrals. For brevity, we only present formal proofs for the discrete case. Additionally, as our primary motivating domain lies within healthcare, we use the term "treatment" in place of "action".

of certainty $\lambda \in [0, 1]$, then π must avoid selecting a at s with the same level of certainty:

$$P_D(s, a) + F_D(s, a) \geq \lambda \implies \pi(s, a) \leq 1 - \lambda. \tag{1}$$

E.g., if a treatment leads to a dead-end or termination with probability more than 80%, then that treatment should be selected for administration no more than 20% of the time. While we would like (1) to hold for the maximum λ, inferring such maximal values is intractable for all state-treatment pairs. Moreover, directly computing P_D and F_D would require explicit knowledge of all dead-end and negative terminal states as well as all transition probabilities for future states. These make the application of (1) nearly impossible. We next develop a learning paradigm to enable (1) from data.

3.4 Dead-end Discovery (DeD)

In order to identify and confirm the existence of dead-end states, we construct two Markov Decision Processes (MDPs) $\mathcal{M}_D$ and $\mathcal{M}_R$ to be identical to $\mathcal{M}$, setting $\gamma = 1$. We also define the following reward functions: $\mathcal{M}_D$ returns -1 with any transition to a negative terminal state (and zero with all other transitions) whereas $\mathcal{M}_R$ returns $+1$ with any transition to a positive terminal state (zero otherwise). Let Q^*_D, Q^*_R, V^*_D and V^*_R denote the optimal state-treatment and state value functions of $\mathcal{M}_D$ and $\mathcal{M}_R$ respectively. Note that due to the reward functions of these MDPs, Q^*_D will always be non-positive while Q^*_R will be non-negative.

Having selected treatment a at state s, using the Bellman equation, we prove[2] that

$$-Q^*_D(s, a) = P_D(s, a) + F_D(s, a) + M_D(s, a) \tag{2}$$

In addition to the quantities defined previously, $M_D(s, a)$ denotes the probability of circumstances in stochastic environments where a negative terminal state ultimately occurs despite receiving optimal treatments at all steps in the future. Equation (2) therefore reveals that $-Q^*_D$ carries special physical meaning: it corresponds to the *minimum probability of a negative outcome*, because future treatments may not necessarily be optimal. Equivalently, $1 + Q^*_D(s, a)$ can be seen as the *maximum hope of a positive outcome* if treatment a is administered at state s.

Building from Fatemi et al. [44], we show that V^*_D of all dead-end states will be precisely -1. By extension, $Q^*_D(s, a) = -1$ for all treatments a at state s if and only if s is a dead-end. In fact, $1 + Q^*_D(s, a)$ provides an appropriate threshold to secure any given policy $\pi(s, a)$. More formally, the following equation guarantees treatment security as presented in (1) for all values of λ:

$$\pi(s, a) \leq 1 + Q^*_D(s, a) \tag{3}$$

In short, for treatment security it is sufficient to abide by the maximum hope of a positive outcome. This construction directly connects RL concept of value functions to the dead-end discovery. While $V^*_D(s)$ enables detecting dead-end states, (3) leverages Q^*_D for treatment avoidance. We establish parallel results for rescue states similarly. The following theorem summarizes the theory and shapes the basis of DeD. See Appendix A1 for the proof and further details.

Theorem 1. Let treatment a be administered at state s, and $P_D(s, a)$ and $P_R(s, a)$ denote the probability of transitioning to a dead-end or rescue state, respectively. Similarly, let $F_D(s, a)$ and $F_R(s, a)$ denote the probability of transitioning to either a negative or positive terminal state. The following hold:

T1 $P_D(s, a) + F_D(s, a) = 1$ if and only if $Q^*_D(s, a) = -1$.

T2 $P_R(s, a) + F_R(s, a) = 1$ if and only if $Q^*_R(s, a) = 1$.

T3 There exists a threshold $\delta_D \in (-1,\ 0)$ independent of states and treatments, such that $Q^*_D(s, a) \geq \delta_D$ for all s and a, unless $P_D(s, a) + F_D(s, a) = 1$.

T4 There exists a threshold $\delta_R \in (0,\ 1)$ independent of states and treatments, such that $Q^*_R(s, a) \leq \delta_R$ for all s and a, unless $P_R(s, a) + F_R(s, a) = 1$.

T5 For any policy π, state s, and treatment a, if $\pi(s, a) \leq 1 + Q^*_D(s, a)$ and $\lambda \in [0, 1]$ exists such that $P_D(s, a) + F_D(s, a) \geq \lambda$, then $\pi(s, a) \leq 1 - \lambda$.

T6 For any policy π, state s, and treatment a, if $\pi(s, a) \geq Q^*_R(s, a)$ and $\lambda \in [0, 1]$ exists such that $P_R(s, a) + F_R(s, a) \geq \lambda$, then $\pi(s, a) \geq \lambda$.

[2] All proofs to the theoretical claims presented in this paper can be found in Appendix A1

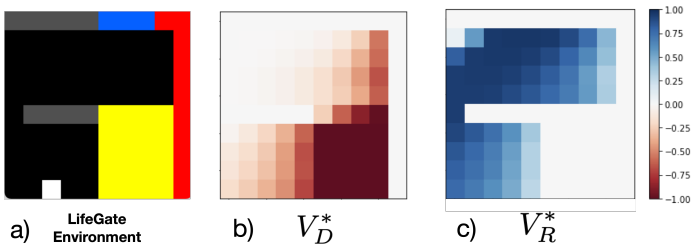

Figure 1: **The Life-Gate Example.** The tabular navigation task of life-gate is illustrated in (**a**). Corresponding dead-end and rescue state-value functions, V_D and V_R, are shown in (**b**) and (**c**), resp. The value functions are learned through Q-learning and with the definition of $\mathcal{M}_D$ and $\mathcal{M}_R$.

It is immediate from (T1) and (T2) that Q_D^* and Q_R^* incorporate complete information when transitioning to a dead-end state or to a rescue state as a result of administrating treatment a at s. (T3) assures that a threshold δ_D exists to separate treatments that lead immediately to dead-ends from alternatives. (T4) allows us to confirm a dead-end by examining if Q_R^* is also smaller than some threshold δ_R. No dead-end can violate δ_R due to (T4) and such a threshold exists. If Q_D^* is available and δ_D is known, then this step is redundant. However, without access to Q_D^* and an accurate δ_D, (T4) helps to confirm any presumed dead-end. Finally, (T5) provides the means by which the treatment policy is guided to avoid dangerous treatments. (T6) is used to also confirm whether the treatment should be avoided. We explain how to practically select the thresholds δ_D and δ_R in Sec. 4.

3.5 Neural Network Based State Construction and Identification

State construction (SC-Network). In domains where solitary observations do not carry salient information for learning the decision-making process, states may need to be constructed from data using a neural network. In these circumstances a separate SC-Network can be used to transform a single or possible sequence of observations into a fixed embedding, considered the state s at time t.

Identification (D-Network and R-Network). In order to approximate Q_D^* and Q_R^*, two separate neural networks can be used to compute Q_D and Q_R for all treatments given a state constructed by the SC-Network. With trained Q_D and Q_R networks, we can then apply thresholds δ_D and δ_R as specified in Theorem 1. As data is limited and non-exploratory, approximation error is inevitable. To mitigate this limitation, the method's sensitivity can be adjusted by adapting the thresholds δ_D and δ_R (additionally, see Proposition 1 and Remarks 1-4 in Appendix A1). Smaller thresholds result in more false negative and less false positive cases. Of note, value-overestimation, a known limitation of deep RL models, will often cause Q_D and Q_R to be larger than Q_D^* and Q_R^* respectively. This naturally reduces false positives while increasing false negatives.

3.6 Toy Problem Validation: Life-Gate

We briefly provide a tabular toy-example (Life-Gate), which involves dead-ends, to empirically illustrates the merit of Theorem 1 by learning Q_D^* and Q_R^* (Figure 1). This toy set-up comprises an interesting case where the agent faces an environment to explore, with no knowledge of possible dangers. Importantly, once a dead-end state (yellow) is reached, it may take some random number of steps before reaching a "death gate" (red). All along such trajectories of dead-end states, the agent still has to choose actions with the (false) hope of reaching a "life-gate" (blue). Discovering any single dead-end state and signaling the agent when the state gets in the scope would be of significant importance. On the other hand, the adjacent states to dead-ends are possibly the most critical ones to alert, as it might be the last chance to still do something (see Appendix A2 for more details).

We next use the tools provided by Theorem 1. The value functions are more than 90% trained, still allowing learning errors. In this example, even with the errors due to lack of full convergence, $\delta_D = -0.7$ and $\delta_R = 0.7$ seem to clearly set the boundary for most states (with a few exceptions due to the errors). If a state is observed whose V_D and V_R values violate these thresholds, the state can be flagged as a dead-end with high probability. Setting a lower threshold can help to raise flags earlier on, when the conditions are becoming high-risk, but it is still not too late. We can apply the same thresholds to further flag high-risk actions (not shown). Lastly, we also remark (T1) from Theorem

1. It can be seen that only for all the yellow area (aside from the few erroneous states), $V_D = -1$. Clearly, no dead-end state can be a rescue, as seen by $V_R = 0$ for the yellow area too.

4 Empirical Setup for Dead-end Analysis

Data: We use DeD to identify medical dead-ends in a cohort of septic patients drawn from the MIMIC (Medical Information Mart for Intensive Care) - III dataset (v1.4) [21, 45]. This cohort totals 19,611 patients (17,730 survivors and 1,881 nonsurvivors), with 47 observation variables, and 25 treatment choices (5 discrete levels for each of IV fluid and vasopressor). We follow prior work [24] and aggregate each variable every four hours using the per-patient variable mean if data is present, or impute using the value from the nearest neighbor.

Terminal States. In our ICU setting, possible terminal states are either patient recovery (discharge from ICU) or death. We define "death" as the last recorded point in the EMR of nonsurviving patients when expiration is imminent, but may not necessarily be the biological point of death. In practice this definition of terminal state may occur hours or days before biological death and covers situations where care support devices are disconnected, when a patient requests a cessation of treatment, etc.

Our goal is to identify all *medical dead-end* states, defined as patient states from which death is unavoidable, regardless of future treatments. Relatedly, we also desire to discover all treatments that may possibly lead to a medical dead-end state in order to learn which treatments to avoid.

SC-Network. As observations of patient health are inherently partial, we need an informative latent representation of state [46], sufficient for evaluating treatment security. To form these state representations we process a sequence of observations prior to and including any time t as well as the last selected treatment to form the state s_t. We train a standalone State Construction (SC) network using Approximate Information State (AIS) [47] in a self-supervised manner for this purpose. Details of AIS and how it is used to train the SC-Network are included in Appendix A3.

D-Network and R-Network. We feed computed SC-Network states to D- and R-networks to approximate Q_D^* and Q_R^*. We use the double-DQN algorithm [48] to train each network (details included in Appendix A4). The outputs of trained D- and R- Networks produce value estimates of both the embedded patient state and all possible treatments to evaluate the probability of transitioning to a dead-end. This process of determining possible high-risk treatments is central to our proposed DeD approach.

Training: We train the SC-, D-, and R- networks in an offline manner, using retrospective data (Fig. A2). All models are trained with 75% of the patient cohort (14,179 survivors, 1,509 nonsurvivors), validated with 5% (890 survivors, 90 nonsurvivors), and we report all results on the remaining held out 20% (2,660 survivors, 282 nonsurvivors). Further details of how the patient cohort is processed are provided in Appendix A5. Finally, to mitigate the data imbalance between surviving and non-surviving patients we use an additional data buffer that contains only those trajectories that include *transitions to death terminal states*. Thus, a stratified minibatch of size 64 is constructed of 62 samples from the main data, augmented with 2 samples from this additional buffer, all selected uniformly. This same minibatch structure is used for training each of the three networks. For the training details see Appendix A3 and A4.

5 Empirical Results

5.1 Septic Dead-End State Prediction

Experiment. In order to flag potential non-secure treatments, we examine if Q_D and Q_R of each treatment at a given state pass certain thresholds δ_D and δ_R, respectively. To flag potential dead-end states, we need to probe the state values, for which we examine the *median* of Q (rather than *max* of Q) against similar thresholds. Using the median helps to avoid extreme approximation error due to generalization from potentially insufficient data. We found that $\delta_D = -0.25$ and $\delta_R = 0.75$ minimize both false positives and false negatives, and use these as the thresholds for raising "red" flags. We also define a second, looser threshold of $\delta_D = -0.15$ and $\delta_R = 0.85$, as raising "yellow" flags with higher sensitivity but increased false positives. This looser threshold targets an early

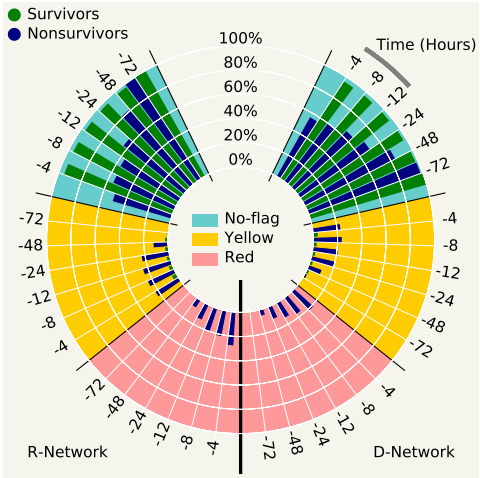

Figure 2: **Flag emergence for ICU patients.** Histograms of median Q according to the flag status, for both surviving (green) and non-surviving patients (blue) according to the R-Network (left) and D-Network (right). The bars are plotted according to the time prior to the recorded terminal state (the maximum trajectory length is 72 hours) and measure the percentage of patients whose states raise either a red, yellow or no flag. There is a clear worsening trend of state values for nonsurviving patients as they approach their terminal state, beginning as early as 48 hours prior.

indication of a patient's health condition deteriorating toward a dead-end state. In Appendix Fig. A5 we report histogram of values at different quantiles, from which we established these thresholds.

Results. Using the specified thresholds, DeD identifies increasing percentages of patients raising fatal flags as nonsurvivors approach death (Figure 2 and Appendix Table A3). Note the distinctive difference between the trend of values in survivors (green bars) and nonsurvivors (blue bars). Over the course of 72 hours in the ICU, survivor trajectories raise nearly no red flag for both networks. In contrast, nonsurvivor trajectories demonstrate a steep reduction in *no-flag* zone with increasing numbers of patients flagged in the *Red* zone. The *Yellow* zone is dominated largely by the nonsurvivors, yet there are also survivors who ultimately recover. Under the red-flag threshold, more than 11% of treatments administered to non-surviving patients are identified to be detrimental 24 hours prior to death with a 0.7% false positive rate (Appendix Table A3). We further identify that 3.8% of non-surviving patient cases have entered unavoidable dead-end trajectories up to 48 hours before recorded expiration, with only a 0.2% false positive rate, i.e., patients misidentified as near death.

We find that 4.7% of nonsurviving patients maintain the red flag for their last 24 hours recorded in the ICU before reaching a death terminal state. This monotonically increases to 12.1% for patients who maintain a red flag through their final 8 hours of care (Appendix Fig. A6**b**,**c**). These patients likely reached a dead-end with no subsequent chance of recovery; this is as compared to 91.1% of nonsurviving test patients with no flag raised in their first 8 hours (Appendix Fig. A6**d**).

There is a distinct difference between remaining on a flag for survivors and nonsurvivors (Appendix Fig. A6**a**). Even with our red threshold, very few survivors (0.7%) raise and remain on red-flag for more than eight hours, decreasing to nearly zero for longer periods. In contrast, 24.6% of nonsurvivors remain on red flags for similar duration with a fat tail. These results suggest that red-flag membership for long periods strongly correlates with mortality, inline with our theoretical analysis.

5.2 First Flag Analysis

Experiment. To further support our hypothesis that dead-end states exist in sepsis and may be preventable, we align patients according to the point in their care when a flag is "first raised". We select all trajectories in the test data with at least 24 hours (6 steps) prior to the first flag and at least 16 hours (4 steps) afterwards (77 surviving and 74 nonsurviving patients). This window excludes patients with flags that occur either too early or too late. This allows for an investigation of the average trend of patient observations, administered treatments as well as the measures used in DeD over a sufficiently large window (Figure 3).

Results. The V and Q values estimated by DeD have similar behavior in survivors and nonsurvivors prior to the first flag, but values diverge after the flag is raised. Notably, the time step pinpointed by DeD to raise a flag corresponds to a similar diverging trend among various clinical measures, including SOFA and patient vitals (Figure 3**a**,**b**). This distinct behavior is also seen if looser threshold values are used for δ_D and δ_R (Appendix Fig. A8). After the flag is raised there is slight improvement in all

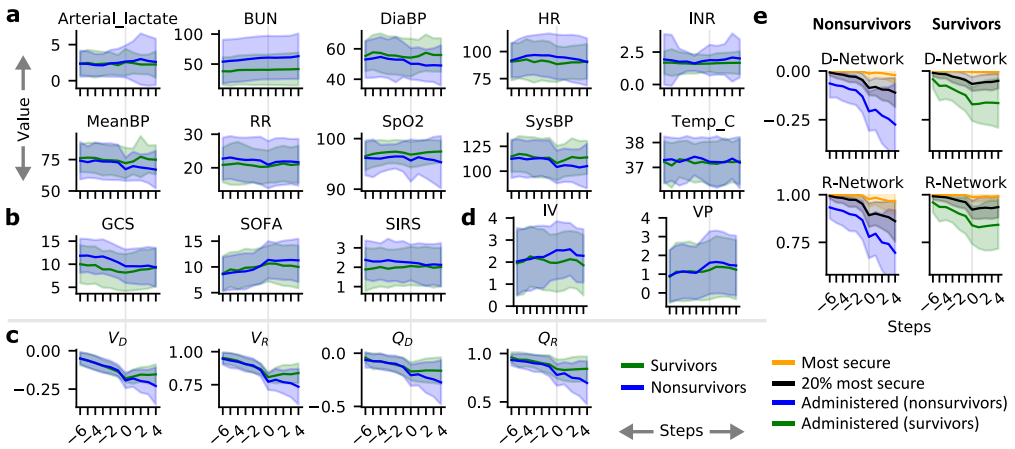

Figure 3: **Trend of measures around the first raised flag.** Various measures are shown 24 hours (6 steps) before the first (red or yellow) flag is raised and 16 hours (4 steps) afterward. All nonsurviving (blue) and surviving (green) patient trajectories that fall within this window are averaged, shaded areas represent a single standard deviation. (**a**) selected key vital measures and lab tests, (**b**) established clinical measures, and (**c**) DeD value measures of state (V) and administered treatment (Q) from the D- and R-Networks and, (**d**) administered treatments. There is a clear turning point 4 to 8 hours prior to the flag being raised, which precisely corresponds to a drastic increase of VP and IV treatments. (**e**) the value of the maximum, the 5th maximum (20% best) and the actually administered treatment, demonstrating that better treatments were available when the chosen treatments were administered.

value estimates, perhaps in response to the change in treatment. However the values of nonsurviving patient trajectories quickly collapse while survivors continue to improve.

The results of this analysis suggest two main points. First, DeD identifies a clear critical point in the care timeline where nonsurviving patients experiencing an irreversible deterioration in health. Second, there is a significant gap (Figure 3**e**) between the value of administered treatments and the 20% most secure ones (5 out of 25). The critical point appears to arise when a patient's condition shifts towards improvement or otherwise enters a dead-end towards expiration. Perhaps most notable is that there is a clear inflection in the estimated values 4 to 8 hours prior to a flag being raised. Signaling this shift in the inferred patient response to treatment and the resulting flag may be used to provide an early indicator for clinicians (more conservative thresholds may be used to signal earlier). The trend of survivors shows that there is still hope to save the patient at this point. Note that all these patients are quite similar in terms of both D/R values and their SOFA score prior to this point. SOFA does appear correlated with DeD, but this is not always the case at the individual level. The trend of value functions in a patient can be significantly more aggressive than SOFA (see Fig. A9 – Fig. A12). Hence, SOFA alone is not sufficient for dead-end identification. DeD is however a provable methodology to this end. Figure 3**d** advocates that the choice of treatment may play a role in entering dead-ends, since the divergence occurs before the flag. The gap in value between the administered treatments and those with the highest estimated security suggest that better treatments were available, even for patients who eventually recover (Figure 3**e**).

5.3 Individual Trajectories

Experiment. In our final analysis we extract relevant information surrounding a patient's value estimates from the electronic health record data, including the recorded clinical notes. We also use t-SNE [49] to project the state representations of the patient's trajectory, embedded using the SC-Network, among all recorded states in the test data.

Results. The clinicians' chart notes confirm existence of dead-ends with a noted need for intubation, hypotension, and a discussion of moving the patient's care to "comfort measures only" (Figure 4**b**). Moreover, certain areas in the t-SNE projection of observed patient states appear to correspond with dead-end states (Figure 4**c**,**d**). Notably, the dramatic shift of clinically established measures such as SOFA and GCS closely follow the decrease in DeD estimated values (Figure 4**a**). This is similar to

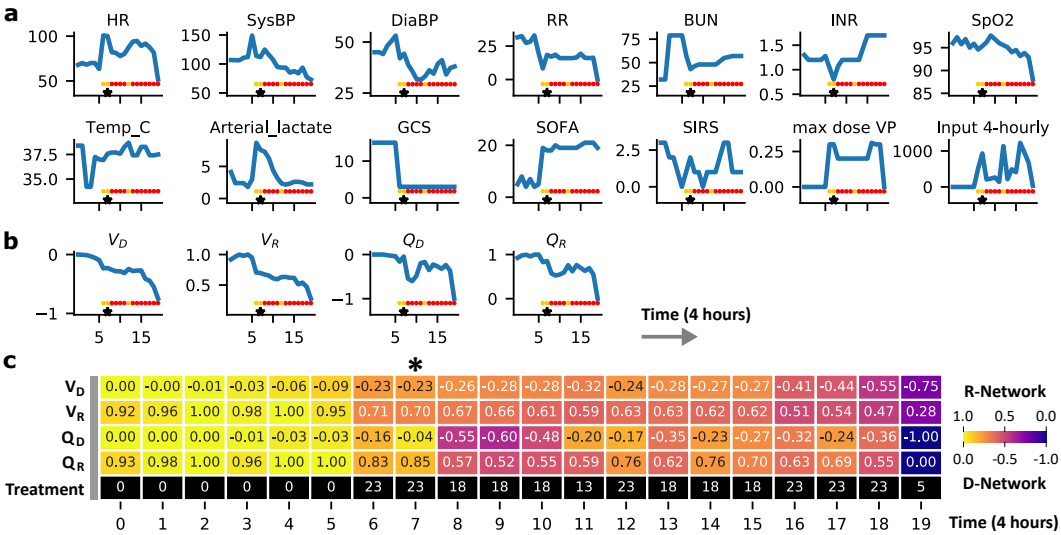

Figure 4: **Events at ICU.** Certain vitals (**a**) and DeD value measures from both D- and R-Networks (**b**) are shown for a non-surviving patient (ICU-Stay-ID 262011). The black asterisks demonstrate the presumed onset of sepsis at step 7, and the color dots corresponds to the raised red or yellow flags. Lastly, (**c**) illustrates steps along the patient's trajectory with DeD estimated values and selected treatments. Notably, from steps 5 to 6 the state has a sudden jump to a low-value region that it fails to escape from, aligned with significant inflections in the recorded vitals, approximately 4 to 8 hours before presumed onset of sepsis. (See Appendix Fig. A9 – Fig. A12 for additional data and patients)

the trends seen among both nonsurvivor and survivor patients prior to raised flags (Figure 3). This qualitative analysis suggests that the estimates of Q_D and Q_R are reliable and informative, supporting our prior conclusions. Additional nonsurvivor patients are presented in Appendix Fig. A9–Fig. A12.

6 Discussion

In this work, we have introduced an RL-based approach for learning what treatments *to avoid* based on observed patterns in limited offline data. We target avoiding treatments proportional to their chance of leading to dead-ends, regions of the state space from which negative outcomes are inevitable. We establish theoretical results that expand the concept of dead-ends in RL, facilitating the notification of high-risk treatments or, as applied to healthcare, septic patient conditions with increased likelihood of leading to a dead-end. Globally, sepsis is a leading cause of mortality [26, 50, 51], and improved efficacy of treatment could have a significant impact both in terms of saving lives and reducing costs.

Our work lays the groundwork for dead-end analysis in medical settings and is, to the best of our knowledge, the first use of RL to flag bad treatments rather than finding the best ones through estimating an optimal policy π^*. Our algorithm is generic, using RL methodology that is formally guaranteed to hold the security condition re-established in this paper. The discovery of dead-end states, and the treatments that likely lead to them, provides actionable insights in intensive care intervention. Further improvement of DeD's prediction quality could target additional features from the EMR environment, such as pre-ICU admission co-morbidities. In future work, we also hope to explore the specific drugs and dosages used in treatment.

Given its general construction, DeD is well matched for safety-critical applications of RL in data-constrained settings where it may be too expensive or unethical to collect additional exploratory data. With formal guarantees of satisfying the security condition, DeD is suitable for broader adoption when developing critical insights from retrospective data. Our framework is particularly relevant to data-constraint offline RL application domains such as robotics, industrial control, and automated dialogue generation where negative outcomes can be clearly identified [52].

Data and Code Availability

We intend to release our code and pre-trained neural models. The MIMIC-III databases (DOI: 10.1038/sdata.2016.35) that support the findings of this study are publicly available through Physionet website: https://mimic.physionet.org, which facilitates reproducibility of the presented results.

References

[1] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998. ISBN 0262193981.

[2] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[3] Travis Mandel, Yun-En Liu, Sergey Levine, Emma Brunskill, and Zoran Popovic. Offline policy evaluation across representations with applications to educational games. In *AAMAS*, pages 1077–1084, 2014.

[4] Susan A Murphy, Mark J van der Laan, James M Robins, and Conduct Problems Prevention Research Group. Marginal mean models for dynamic regimes. *Journal of the American Statistical Association*, 96(456):1410–1423, 2001.

[5] 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.

[6] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[7] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1407–1416. PMLR, 10–15 Jul 2018. URL `http://proceedings.mlr.press/v80/espeholt18a.html`.

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

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

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

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

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

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

[14] Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pages 104–114. PMLR, 2020.

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

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

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

[18] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Advances in Neural Information Processing Systems*, pages 11784–11794, 2019.

[19] Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

[20] Ziyu Wang, Alexander Novikov, Konrad Żołna, Jost Tobias Springenberg, Scott Reed, Bobak Shahriari, Noah Siegel, Josh Merel, Caglar Gulcehre, Nicolas Heess, et al. Critic regularized regression. In *Advances in Neural Information Processing Systems*, 2020.

[21] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, 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.

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

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

[24] 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.

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

[26] Clifford S. Deutschman and Kevin J. Tracey. Sepsis: Current dogma and new perspectives. *Immunity*, 40(4):463 – 475, 2014. ISSN 1074-7613. doi: https://doi.org/10.1016/j.immuni.2014.04.001. URL `http://www.sciencedirect.com/science/article/pii/S1074761314001150`.

[27] Mervyn Singer, Clifford S. Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R. Bernard, Jean-Daniel Chiche, Craig M. Coopersmith, Richard S. Hotchkiss, Mitchell M. Levy, John C. Marshall, Greg S. Martin, Steven M. Opal, Gordon D. Rubenfeld, Tom van der Poll, Jean-Louis Vincent, and Derek C. Angus. The third international consensus definitions for sepsis and septic shock (sepsis-3). *JAMA*, 315(8):801, feb 2016. doi: 10.1001/jama.2016.0287. URL `https://doi.org/10.1001%2Fjama.2016.0287`.

[28] Jean-Louis Vincent, Steven M Opal, John C Marshall, and Kevin J Tracey. Sepsis definitions: time for change. *The Lancet*, 381(9868):774 – 775, 2013. ISSN 0140-6736. doi: https://doi.org/10.1016/S0140-6736(12)61815-7. URL `http://www.sciencedirect.com/science/article/pii/S0140673612618157`.

[29] C Torio and B Moore. National inpatient hospital costs: The most expensive conditions by payer, 2013. Statistical Brief #204. *Healthcare Cost and Utilization Project (HCUP) Statistical Briefs*, May 2016. URL `http://www.hcup-us.ahrq.gov/reports/statbriefs/sb204-Most-Expensive-Hospital-Conditions.pdf`.

[30] Greg S Martin. Sepsis, severe sepsis and septic shock: changes in incidence, pathogens and outcomes. *Expert review of anti-infective therapy*, 10(6):701–706, 2012.

[31] P. E. Marik, W. T. Linde-Zwirble, E. A. Bittner, J. Sahatjian, and D. Hansell. Fluid administration in severe sepsis and septic shock, patterns and outcomes: an analysis of a large national database. *Intensive Care Med*, 43(5):625–632, May 2017.

[32] J. Waechter, A. Kumar, S. E. Lapinsky, J. Marshall, P. Dodek, Y. Arabi, J. E. Parrillo, R. P. Dellinger, A. Garland, S. Dial, P. Ellis, D. Feinstein, D. Gurka, J. Guzman, S. Keenan, A. Kramer, A. Kumar, D. Laporta, K. Laupland, B. Light, D. Maki, G. Martinka, Z. Memish, Y. Mirzanejad, G. Patel, C. Penner, D. Roberts, J. Ronald, D. Simon, S. Sharma, N. A. Shirawi, Y. Skrobik, G. Wood, K. E. Wood, S. Zanotti, M. W. Ahsan, M. Bahrainian, R. Bohmeier, L. Carter, H. Chou, S. Delgra, C. Egbujuo, W. Fu, C. Gonzales, H. Gulati, E. Halmarson, J. Hansen, Z. Haque, J. Harvey, F. Khan, L. Kolesar, L. Kravetsky, R. Kumar, N. Merali, S. Muggaberg, H. Paulin, C. Peters, J. Richards, C. Schorr, H. Serrano, M. Suleman, A. Singh, K. Sullivan, R. Suppes, L. Taiberg, R. Tchokonte, O. A. Torshizi, and K. Wiebe. Interaction between fluids and vasoactive agents on mortality in septic shock: a multicenter, observational study. *Crit. Care Med.*, 42(10):2158–2168, Oct 2014.

[33] Robert Gauer, Damon Forbes, and Nathan Boyer. Sepsis: diagnosis and management. *American family physician*, 101(7):409–418, 2020.

[34] Aniruddh Raghu, Matthieu Komorowski, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Continuous state-space models for optimal sepsis treatment: a deep reinforcement learning approach. In *Machine Learning for Healthcare Conference*, pages 147–163. PMLR, 2017.

[35] 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.

[36] 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.

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

[38] Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

[39] Philip S Thomas. *Safe reinforcement learning*. PhD thesis, University of Massachusetts Libraries, 2015.

[40] Dylan Hadfield-Menell, Anca Dragan, Pieter Abbeel, and Stuart Russell. Cooperative inverse reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 3916–3924, 2016.

[41] Dylan Hadfield-Menell, Smitha Milli, Pieter Abbeel, Stuart Russell, and Anca D Dragan. Inverse reward design. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6768–6777, 2017.

[42] Philip S Thomas, Bruno Castro da Silva, Andrew G Barto, Stephen Giguere, Yuriy Brun, and Emma Brunskill. Preventing undesirable behavior of intelligent machines. *Science*, 366(6468): 999–1004, 2019.

[43] Romain Laroche, Paul Trichelair, and Remi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *International Conference on Machine Learning (ICML)*, June 2019.

[44] Mehdi Fatemi, Shikhar Sharma, Harm Van Seijen, and Samira Ebrahimi Kahou. Dead-ends and secure exploration in reinforcement learning. In *International Conference on Machine Learning*, pages 1873–1881, 2019.

[45] Alistair Ew Johnson, David J Stone, Leo A Celi, and Tom J Pollard. The MIMIC code repository: enabling reproducibility in critical care research. *J. Am. Med. Inform. Assoc.*, 25(1):32–39, January 2018.

[46] 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. URL `http://proceedings.mlr.press/v136/killian20a`.

[47] Jayakumar Subramanian and Aditya Mahajan. Approximate information state for partially observed systems. In *Conference of Decision and Control (CDC), Nice, France*, 2019.

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

[49] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[50] Jean-Louis Vincent, John C Marshall, Silvio A Ñamendys Silva, Bruno François, Ignacio Martin-Loeches, Jeffrey Lipman, Konrad Reinhart, Massimo Antonelli, Peter Pickkers, Hassane Njimi, Edgar Jimenez, and Yasser Sakr. Assessment of the worldwide burden of critical illness: the intensive care over nations (icon) audit. *The Lancet Respiratory Medicine*, 2(5):380 – 386, 2014. ISSN 2213-2600. doi: https://doi.org/10.1016/S2213-2600(14)70061-X. URL `http://www.sciencedirect.com/science/article/pii/S221326001470061X`.

[51] C. Fleischmann, A. Scherag, N. K. Adhikari, C. S. Hartog, T. Tsaganos, P. Schlattmann, D. C. Angus, and K. Reinhart. Assessment of Global Incidence and Mortality of Hospital-treated Sepsis. Current Estimates and Limitations. *Am. J. Respir. Crit. Care Med.*, 193(3):259–272, Feb 2016.

[52] 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.

[53] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[54] Matthieu Komorowski. AI Clinician. `https://github.com/matthieukomorowski/AI_Clinician`, 2018. Accessed: 2019-08-16.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Section 3.5 and 4
   (c) Did you discuss any potential negative societal impacts of your work? [No]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] Appendix A1

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [No] We detailed the data and will release code if the paper is accepted.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Section 4 and Appendix A2-A4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Appendix Figures
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] Not important in our case
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] MIMIC data, Section 4
   (b) Did you mention the license of the assets? [No]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]