To Reduce Healthcare Workload: Identify Critical Sepsis Progression Moments through Deep Reinforcement Learning

Song Ju
Computer Science
North Caroline State University
sju2@ncsu.edu

Yeo Jin Kim
Computer Science
North Caroline State University
ykim32@ncsu.edu

Markel Sanz Ausin
Computer Science
North Caroline State University
msanzau@ncsu.edu

Maria Mayorga
Computer Science
North Caroline State University
memayorg@ncsu.edu

Min Chi
Computer Science
North Caroline State University
mchi@ncsu.edu

***Abstract*—Healthcare systems are struggling with increasing workloads that adversely affect quality of care and patient outcomes. When clinical practitioners have to make countless medical decisions, they are not always able to make them prudently or consistently. In this work, we formulate clinical decision making as a reinforcement learning (RL) problem and propose a *human-controlled machine-assisted (HC-MA)* decision making framework whereby we can simultaneously give clinical practitioners (the humans) control over the decision-making process while supporting effective decision-making. In our HC-MA framework, the role of the RL agent is to *nudge* clinicians *only if they make suboptimal decisions at critical moments*. This framework is supported by a general Critical Deep RL (Critical-DRL) approach, which uses Long-Short Term Rewards (LSTRs) and Critical Deep Q-learning Networks (CriQNs). Critical-DRL's effectiveness has been evaluated in both a GridWorld game and real-world datasets from two medical systems: Christiana Care Health System (CCHS) in Newark, Delaware and Mayo Clinic in Rochester, Minnesota, USA for septic patient treatment. We found that our Critical-DRL approach, by which decisions are made at critical junctures, is as effective as a fully executed DRL policy and moreover, it enables us to identify the critical moments in the septic treatment process, thus greatly reducing burden on medical decision-makers by allowing them to make critical clinical decisions without negatively impacting outcomes.**

***Index Terms*—Reinforcement Learning, Sepsis, Critical Decision**

I. Introduction

Many studies show that the medical operational features of the healthcare delivery environment impact the quality of care and hence patient health outcomes [1]–[5]. For example, nurse workload, which is a function of the number and complexity of patients a nurse cares for [3], has been shown to affect length of stay (LOS) and rates of hospital acquired infections [6], [7]. Workload is defined as "the task demand of accomplishing mission requirements for the human operator" [8]. Interpretation and quantification of workload in healthcare delivery depends on many factors [9] and has been quantified using objective, physiological and subjective measures [10]. Treatment decision-making is a type of workload which plays a critical role in clinician performance. In hospitals, physicians make a large number of clinical decisions from defining the problem, to evaluating test results, to treatments. For example, in one study, an average of 13.4 decisions were made during a patient visit [11]. However, clinical decision-makers do not always make optimal or consistent decisions in such complex tasks for many reasons. One of them is "decision fatigue". After a long series of decisions, people tend to develop cognitive fatigue which can lead them to favor the seemingly easiest option over all the others. One study found that decision-makers tend to procrastinate, be less persistent, and even fail to recognize decision opportunities at all [12]. For instance, nurses who suffer from decision fatigue are more likely to make conservative decisions, which indicates that they prefer to choose the 'default' option [13]. An abundance of conservative decisions can have unwanted consequences in resource and time-limited environments.

Like many real-world tasks, decision-making in healthcare can present itself as a sequential multi-step learning problem when the outcome of the selected actions is delayed. Reinforcement Learning (RL) offers an effective data-driven solution based on a mathematically-grounded framework that learns an optimal policy from data to maximize expected reward [14]. In particular, Deep RL (DRL) effectively models high-dimensional data and has been applied to healthcare [15], [16]. In real-world domains like healthcare, however, such automated decision-making approaches are undesirable and unacceptable due to ethical, legal, and moral reasons.

We propose a general **human-controlled machine-assisted (HC-MA) DRL framework** that nudges decision-makers towards promising medical treatment decisions. As shown in Fig 1, in our HC-MA framework, the human (medical

practitioners such as physicians or clinician teams) is the front-end decision-maker, whereas the RL agent provides back-end support. In Fig 1, there is only one physical environment which is modelled differently S_{human} for the human and S_{RL} for the RL agent. It is because the way humans perceive the environment differs greatly from the way RL agents model it, and perceptions of people within the same physical environment can also vary greatly. Consequently, human's decision a_{human} can be different from the RL agent's decision a_{RL}. With this HC-MA framework, the role of the RL agent is to support the human to make effective decisions that contribute to the desired outcomes and to prevent pitfalls by nudging the human to make optimal decisions only if the human makes suboptimal decisions when it matters, that is, ***at critical moments***.

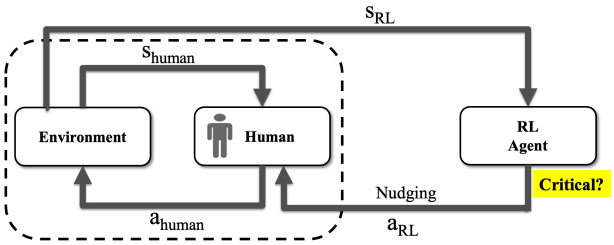


Fig. 1. Human-Controlled Machine-Assisted Framework

The ultimate goal of this HC-MA framework is to strike a balance between giving human decision-makers control over the process while supporting effective decision-making. In this work, we focus on an extremely challenging task: sepsis treatment. Sepsis, defined as infection plus systemic manifestations of infection, is the greatest in-hospital cause of mortality and source of expense; the syndrome has a high mortality (43.8% in the high-target group and 42.3% in the low-target group at 90 days, where the high-target group is a single-center trial targeting high MAP, which is the driving pressure of tissue perfusion [17]) even if treated according to recommended guidelines [18]. This is due, in part, to difficulties in diagnosis and delayed treatment. For every one hour delay in treatment of severe sepsis/shock with antibiotics, there is 10% decrease in patient survival probability [19]. On the other hand, there are many barriers to timely, effective treatment of sepsis: response and treatment depend on many factors including the type of infection and the predisposition. The treatment of sepsis patients is complex – the patient's condition is stochastic and dynamically changing during the diagnosis process. Furthermore, the diagnosis of sepsis requires the selection and ordering of potentially invasive and/or costly imprecise tests. Patients' responses to treatment are uncertain, and the treatment itself is continually evolving as the care provider gains insight into the patient's condition and learns more about the patient's vital signs, laboratory tests, and their response to treatment over time.

Similar to a large body of real-world tasks, sepsis treatment can be characterized as a temporal sequential multi-step decision process, where the outcome of the selected treatment is often delayed. To identify critical moments, we proposed and developed a *Critical Deep RL (Critical-DRL)* framework based on the Long-Short Term Rewards (LSTRs) and Critical Deep Q-Network (CriQN) algorithms for inducing the critical policy. In the critical policy, optimal actions must be taken in critical states and any action is allowed to be taken in non-critical states. Critical-DRL's effectiveness is assessed first on a GridWorld, and then on two medical systems: Christiana Care Health System (CCHS) in Newark, Delaware and Mayo Clinic in Rochester, Minnesota, USA on the task of sepsis shock prevention. We evaluated our proposed Critical-DRL framework from two aspects: 1) effectiveness in that the critical policy should be as effective as a fully-executed policy 2) how much it honors HC-MA decision making. Our results show that the proposed Critical-DRL framework does indeed identify critical decisions, in that critical policy can be implemented as effectively as fully implemented policies. Moreover, it can be leveraged in our HC-MA framework to lower burden on medical decision-makers by allowing them to make critical clinical decisions without compromising outcomes. Our **main contributions** are: 1) we developed a Critical-DRL framework to identify critical decisions and evaluated it; 2) we proposed a HC-MA framework based on Critical-DQN and investigated its potential for reducing healthcare workload.

II. Method

Our Critical-DRL framework is an offline approach. RL approaches have been categorized as being either online or offline. In the online RL, the agent learns while interacting with the environment; in the offline case, the agent learns the policy from pre-collected data. Online RL algorithms are generally appropriate for domains where interacting with simulations and actual environments is computationally cheap and feasible. Simulations in healthcare domains can be especially difficult due to disease progression modelling being very complex, poorly understood processes; learning policies while working with patients is unethical, if not illegal. Therefore, we focus on offline RL. Inspired by neuroscience [20], our Critical-DRL framework is built upon the Long-Short Term Rewards, which are heuristics to measure the importance of a state. Throughout the following sections, we will describe the Long-Short Term Rewards approach for defining critical states, our Critical Deep Q-Network algorithm, and the critical policies for evaluating the quality of critical decisions.

A. Long Term Rewards

In conventional RL, an agent's interactions with an environment are often framed as a Markov Decision Process (MDP), where at each time-step the agent observes the environment in state s, it takes an action a and receives a scalar reward r and the environment moves to the next state. $Q(s, a)$ is defined as the expected cumulative rewards the agent will receive by taking action a at state s and following the policy to the end. Much of prior research applied Q-value difference among different actions on a state as a heuristic value to measure the importance of the state [21]–[24]. In general, the higher the

difference, the more important the state should be. Intuitively, if all the actions for a given state have the same Q-value, then it does not matter which action should be taken because all the actions will lead to the same ultimate outcome. On the other hand, if the difference of Q-values among different actions is large, then taking a suboptimal action can result in a significant loss in the final outcome. Therefore, we define the ***Long Term Reward (LongTR)*** as the difference between the cumulative future rewards of the best action and that of the worst action for a given state s expressed as:

$$LongTR(s) = \max_{a} Q(s,a) - \min_{a'} Q(s,a') \tag{1}$$

B. Short-Term Rewards

As part of making decisions, humans use immediate rewards as well as long-term rewards [20]. It is often said that humans prefer immediate rewards to long-term rewards, particularly when the immediate rewards are large. This is due to the fact that the real world is unpredictable and dynamic. In general, the longer someone waits for a future return, the higher the risk of losing it. So while we use Q-values for long term reward, we use immediate rewards for ***Short Term Reward (ShortTR)***.

However, in many real-world applications like healthcare, the rewards are often delayed until the end of the trajectory. Different from the delayed rewards in the classic mouse-in-the-maze situations where agents receive insignificant rewards along the path and a significant reward in the final goal state (the food), in healthcare, there are immediate rewards along the way but they are often *unobservable*. This is due to the nature of disease, which makes it difficult to assess patient's health moment by moment. For instance, the most proper rewards in healthcare is the patient outcomes, which are often unavailable until the end of the trajectory. Therefore, the challenge is how to infer these unobservable immediate rewards from the delayed rewards, while taking the noise and uncertainty in the data into account.

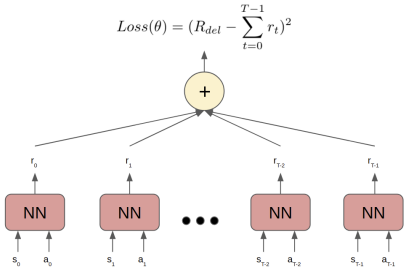


Fig. 2. Architecture of InferNet

In this work, following [25] we apply a neural network based approach (InferNet) to infer "immediate" rewards from delayed rewards. Fig 2 shows the general architecture of the InferNet. Basically, given a trajectory $(s_0, a_0, s_1, a_1 \cdots s_{T-1})$ with length T to the InferNet as input, it outputs an "inferred" immediate reward $r_t = f(s_t, a_t|\theta)$ for each state-action pair in the trajectory. Here, θ indicates the parameters (weights and biases) of the neural network. When learning the "inferred" immediate rewards, there should be a constraint that: the sum of all the predicted rewards in one trajectory is equal to the delayed reward, as shown in Equation 2 where R_{del} indicates the delayed reward.

$$R_{del} = f(s_0, a_0|\theta) + f(s_1, a_1|\theta) + ... + f(s_{T-1}, a_{T-1}|\theta) \tag{2}$$

Therefore, the InferNet is trained by minimizing the loss function between the sum of the predicted rewards and the delayed reward for each trajectory, as shown in Equation 3.

$$Loss(\theta) = (R_{del} - \sum_{t=1}^{T} f(s_t, a_t|\theta))^2 \tag{3}$$

Once the InferNet model is trained, we can predict the immediate rewards, the ShortTRs, for any state-action pairs.

C. Critical Decision & Critical Policy

Generally, we refer to **critical decisions** as those which have a significant influence on the desired outcome, whereas non-critical decisions have a lesser impact. While Long-Short Term Rewards defined above can be regarded as heuristics for estimating the relative significance of a decision; however, quantifying the *ratio* of critical vs. non-critical decisions is challenging. Therefore, we categorize decision into binary categories (critical versus non-critical) by varying thresholds based on both long-term and short-term rewards.

Furthermore, our ***Critical Policy*** *is a policy that takes the optimal action for critical states, but random actions for non-critical states*. The intuition is that in critical states, different actions have great impacts on the desired outcomes, and therefore the optimal action should be taken, whereas in non-critical states, any action can be taken. Consequently, the more accurate the critical decisions are, the more effective the Critical Policy will be. To induce effective critical policy, we proposed Critical Deep Q-Network (CriQN).

D. Critical Deep Q-Network (CriQN)

Deep Q-Network (DQN) is one of the most promising approaches that is widely used in areas like robotics and video games [26]. In DQN, the Q functions are estimated based on the Bellman equation that the optimal policy will be followed all the way to the end. For a single (s, a, r, s') tuple, the Bellman Equation can be expressed as:

$$Q(s,a) = r + \gamma \max_{a'} Q(s', a'). \tag{4}$$

where r is the immediate reward for taking action a at state s; γ is the discount factor; and $Q(s', a')$ is the action-value function for taking action a' at the subsequent state s'.

The Critical Policy, in the critical states, will take optimal actions, while in the noncritical states, any action will be considered. As a result, it fundamentally violates the Q-value assumptions in the Bellman equation by failing to take the optimal actions on non-critical states. In order to take into account our critical decisions, we modify the Bellman equation based on whether the state is critical. The core intuition behind the Critical Deep Q-Network (CriDQN) is that an agent should

take the optimal action if a state is critical, else it can choose any action and thus we have:

$$Q(s,a) = \begin{cases} r + \gamma maxQ(s',a') & \text{s' is critical} \\ r + \gamma meanQ(s',a') & \text{s' is non-critical.} \end{cases} \quad (5)$$

The Equation 5 states that when updating Q-values, the maximum Q-value for a state is used if the state is critical, while if the state is not critical, the average of all possible Q-values is used since any action can be taken.

Algorithm 1 presents the pseudo-code for the CriQN. First of all, the InferNet model is trained to infer the immediate rewards from delayed rewards in the training dataset. Then, there are three parameters in the algorithm: $T^{+}_{ShortTR}$ and $T^{-}_{ShortTR}$ are the ShortTR thresholds originating from the elbows of the inferred immediate reward distribution and, T_{LongTR} is the LongTR threshold used to determine the Q-value difference threshold. Lines 7-17 in Algorithm 1 applies the InferNet to predict the maximum and minimum inferred immediate rewards for the next state s'. If the maximum is larger than $T^{+}_{ShortTR}$ or the minimum is smaller than $T^{-}_{ShortTR}$, the next state s' is critical and a label c_i is added to the tuple. Second, it initializes all Q-values using the inferred immediate rewards to avoid the bias of the neural network.

Lines 26-48 show that for each iteration, we will first calculates the Q-value difference for all states. The Q-value difference threshold T_{Δ} is defined as the top T_{LongTR} percent value in the training dataset. It means that T_{LongTR} percent states with higher Q-value difference are critical. In other words, T_{LongTR} is like a calibrated score (e.g. $p50$, $p80$) to determine how many states are critical, and T_{Δ} is the real cutoff on the Q-value difference value. Finally, for each (s, a, r, s', c') tuple, if the Q-value difference of s' is larger than T_{Δ} or it is identified as critical by the immediate rewards $c' == True$, we consider the state s' as critical and its value function is $max_{a'}Q(s', a'; \theta^{-})$; for non-critical states, their value function are defined as $mean_{a'}Q(s', a'; \theta^{-})$.

In summary, Algorithm 1 applies ShortTR to identify one set of critical states and LongTR to identify another set of critical states. The final critical states are the union of the two sets. More specifically, the set of ShortTR is static because the thresholds $T_{ShortTR}$ and InferNet are pre-defined before training. Though, the set of LongTR is dynamic and determined by the RL policy and threshold T_{LongTR}.

Algorithm 1 Pseudocode of CriQN

1: Train and load InferNet model
2: Initialize the training dataset D as (s, a, r, s') tuples.
3: Initialize the Q function with random parameters θ
4: Initialize the target $\hat{Q}$ function with parameters $\theta^{-} = \theta$
5: Set user-defined parameters: $T^{+}_{ShortTR}$, $T^{-}_{ShortTR}$, T_{LongTR}
6:
7: // Initialize critical states based on immediate rewards
8: **for** each (s_i, a_i, r_i, s'_i) in D **do**
9: $r'_{max} = max(InferNet(s'_i, a'))$
10: $r'_{min} = min(InferNet(s'_i, a'))$
11: **if** $r'_{max} > T^{+}_{ShortTR}$ **or** $r'_{min} < T^{-}_{ShortTR}$ **then**
12: $c'_i = True$
13: **else**
14: $c'_i = False$
15: **end if**
16: $D \leftarrow (s_i, a_i, r_i, s'_i, c'_i)$
17: **end for**
18:
19: // Initialize $Q(s, a)$ as immediate reward
20: **for** each $(s_i, a_i, r_i, s'_i, c'_i)$ in D **do**
21: set $y_i = r_i$
22: **end for**
23: Perform gradient descent on $(y_i - Q(s_i, a_i; \theta))^2$
24: Reset $\hat{Q} = Q$
25:
26: // Main training loop
27: **for** iteration k = 1, 2, ... till convergence **do**
28: Initialize empty array Q_{diffs}
29: **for** each $(s_i, a_i, r_i, s'_i, c'_i)$ in D **do**
30: $Q_{diffs} \leftarrow (maxQ(s_i, a'; \theta^{-}) - minQ(s_i, a'; \theta^{-}))$
31: **end for**
32: $T_{\Delta(Q)}$ = top T_{LongTR} percent of Q_{diffs}
33:
34: **for** each $(s_i, a_i, r_i, s'_i, c'_i)$ in D **do**
35: **if** terminal s'_i **then**
36: Set $y_i = r_i$
37: **else**
38: $Q_{diff} = maxQ(s'_i, a'; \theta^{-}) - minQ(s'_i, a'; \theta^{-})$
39: **if** $Q_{diff} > T_{\Delta(Q)}$ **or** $c'_i == True$ **then**
40: Set $y_i = r_i + \gamma max_{a'}Q(s', a'; \theta^{-})$
41: **else**
42: Set $y_i = r_i + \gamma mean_{a'}Q(s', a'; \theta^{-})$
43: **end if**
44: **end if**
45: **end for**
46: Perform gradient descent on $(y_i - Q(s_i, a_i; \theta))^2$
47: Every C steps reset $\hat{Q} = Q$
48: **end for**

E. Identifying & Evaluating Critical Decisions

A standard RL agent would always take the optimal action, applying maximum-Q in Bellman equations to all states, and thus it is guaranteed to converge and lead to an optimal policy. Our critical RL agent, however, only takes the optimal actions on critical states and any actions on other states, taking average-Q on non-critical states and thus it is not guaranteed to converge. On the other hand, our critical policy can be seen as two relative independent steps: *identifying critical states* and *determining the optimal action for critical states*. Therefore, we explored different combinations of Critical-DQN (CriQN) and DQN for these two steps separately. That is, we explored four critical policies denoted in the form of state-action pairs as: *CriQN-CriQN, CriQN-DQN, DQN-DQN, DQN-CriQN*. For example, the CriQN-DQN refers to use the CriQN to identify critical states and DQN to select optimal actions. Consequently, the performance of the critical policy is determined by both factors: the accuracy of critical state identification and the choice of optimal action on the critical state. In the following, we investigate 1) how the two factors affect the performance of the critical policy and 2) how close the critical policy's performance is to a fully-executed policy.

III. GridWorld Game Testbed

A. GridWorld Description

In the GridWorld game, the agent learns an optimal policy to collect as much reward as possible from the start point to the end point. Fig 3 shows our GridWorld environment. The agent starts from the start state (right bottom corner), explores the 2D space and finishes at the end state (left upper corner). There are several walls in the GridWorld which are marked as black blocks. The agent state is simply represented by the X and Y coordinates.

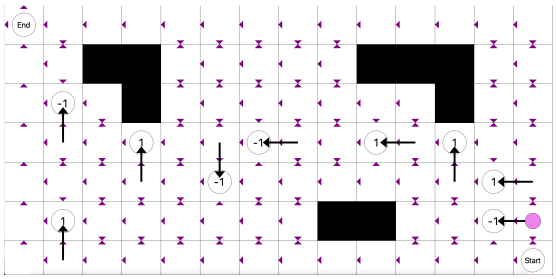


Fig. 3. The Interface of the GridWorld Game

1) Action: There are three actions: up, down and left. In Fig 3, the possible actions for each state are labeled with small purple triangles such that some states have three possible actions whereas others only have two or one possible action(s). The possible action(s) for each state is predefined in the environment, so that the agent never hits the wall or boundary.

2) Reward: In the GridWorld, there is a -0.1 penalty reward for each step, and the agent can collect -1 and +1 rewards. In order to simulate the real world, the reward function is designed in terms of state-action-state, $R(s, a, s')$. The black arrows indicate that if the agent enters the reward state along that arrow, the agent will get the reward; otherwise, the agent will not receive the reward. Furthermore, when the agent enters the reward state, it is forced to move left. This design aims to avoid the agent from collecting the same reward repeatedly without moving towards the terminal state.

3) Stochastic: The GridWorld environment is stochastic in that the same state-action pair can result in a different next state. For example, if the agent takes action 'left', it only has an 85% chance of moving left, and a 15% chance of moving to other possible directions.

B. Experiment Setup

To align with our healthcare application in which online learning is infeasible, we focus on an offline RL approach and follow the three steps: 1) collect the training dataset by random exploration, 2) induce the policies offline and, 3) evaluate the performance of induced policies online.

1) Data Collection: The training dataset contains 1000 trajectories that are generated from a random policy under different random seeds.

2) Offline Learning: Before inducing the policies, we train the InferNet model to infer the immediate rewards in the training dataset. Fig 4 (Left) shows the training process for InferNet that the Root Mean Square Error (RMSE) between the sum of inferred immediate rewards in a trajectory and its delayed reward is decreasing during the training process. After about 1 million training iterations, the InferNet has converged. Fig 4 (Right) shows the inferred immediate rewards distribution in the training dataset. We sorted the inferred immediate rewards in descent order and the X axis shows the ranking percentage for the whole dataset.

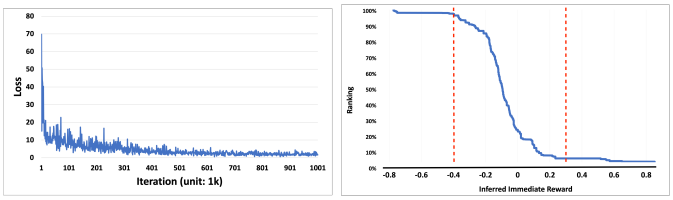


Fig. 4. Left: Training process for InferNet. Right: Inferred Immediate Rewards Distribution

To identify critical states, a key thing is to choose an appropriate threshold that would not include too many trivial decisions, but at the same would not exclude too many critical decisions. For CriQN, the $T^{-}_{ShortTR}$ and $T^{+}_{ShortTR}$ are fixed as -0.4 and $+0.3$ based on the elbows of the inferred immediate reward distribution, which are indicated by the two vertical red dot lines in Fig 4 (Right). For the T_{LongTR} parameter, we cap the critical decisions to be no more than 50% that determines the final outcome and thus, using the training data we explore five different thresholds, the top [10%, 20%, 30%, 40%, 50%] of all the Q-value difference in the training data as the cut-off points for T_{LongTR} rewards and thus five CriQN policies are induced. Therefore, a total of six policies were trained in this step, five CriDQN policies and one DQN policy. More specifically, since we are applying offline learning, the converge criterion is controlled by the number of training iterations. With enough training iterations, both CriDQN and DQN could learn how to act optimally in the GridWorld game.

3) Online Evaluation: In the online evaluation, for any given state, we first apply InferNet to estimate the inferred immediate reward for each action. If the maximum inferred reward is larger than $T^{+}_{ShortTR}$ or the minimum is smaller than $T^{-}_{ShortTR}$, then the state is critical. Second, if the state is not critical, then we utilize the RL policy to calculate Q-value difference and compare with its threshold $T_{\Delta(Q)}$, which is calculated based on the corresponding policy and T_{LongTR} in the training dataset. In the end, if the state is critical, the agent follows the corresponding policy to take the optimal action. Otherwise, the agent can select any action randomly. Thus, there are two stages in the online evaluation, identify critical state and then select optimal action. For example, the critical policy CriQN-DQN will apply CriQN policy to calculate Q-value difference and compare with its threshold to determine critical, but apply DQN policy to select optimal actions.

The performance of the critical policy is measured by the average of cumulative rewards over 100 trials under different random seeds. For a more robust or accurate result, we repeat the entire experiment 20 times with completely different random seeds to minimize the bias caused by data collection.

C. Results

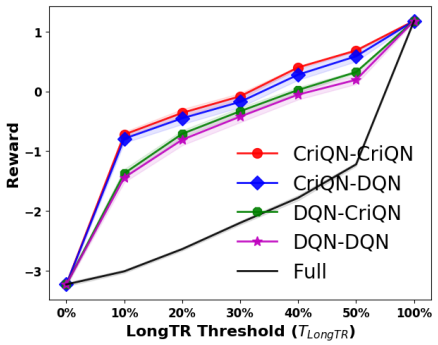


Fig. 5. Online Evaluation Result

Fig 5 shows the online evaluation result. The X axis represents the LongTR threshold used by the corresponding identification policy to identify critical states. It is important to note that we define 100% as a fully-executed DQN policy which carries out the optimal actions all the time, and 0% as a fully random policy which always randomly selects actions. Thus, 0% and 100% indicate the lower and upper performance bounds for critical policy. The Y axis shows the reward (average of 20 replications with the shadows depicting the standard error) received by each critical policy. More specifically, we involved a Full (DQN) policy, which randomly pick states to take optimal actions, as a baseline policy to test whether the identified set of critical states are indeed critical. For example, for the Full policy, $T_{LongTR} = 30\%$ means we randomly pick 30% states to take optimal actions and 70% states to take any actions.

Overall, there is a general trend that the larger the threshold (the more states classified as critical states and take optimal actions), the better the critical policy performs.

1) Performance Comparison: First, we investigate how the identification policy and execution policy may impact the performance of the critical policy. Fig 5 shows that CriQN-CriQN (red) and CriQN-DQN (blue) perform very closely to each other while the performance of DQN-CriQN (green) and DQN-DQN (magenta) are very close; more importantly, the former two outperform the latter two across different LongTR thresholds. It suggests that the CriQN is more accurate in identifying critical states than DQN while for carrying out the optimal actions, both CriQN and DQN can be effective. Second, when comparing the Full policy with the four critical policies, the Full policy performs significantly worse than the others. It means that identified critical states through LSTRs are indeed critical and better than a randomly picked set of states. Finally, when comparing to the fully-executed policy (100% threshold), the CriQN-CriQN and CriQN-DQN with threshold 50% can reach 90% performance of a fully-executed DQN policy. Note that because DQN-CriQN (green) and DQN-DQN (magenta) have very close performance, for simplicity reasons we only include DQN-DQN for the purpose of comparisons in the following healthcare dataset.

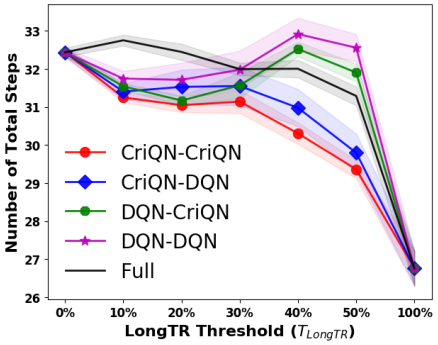
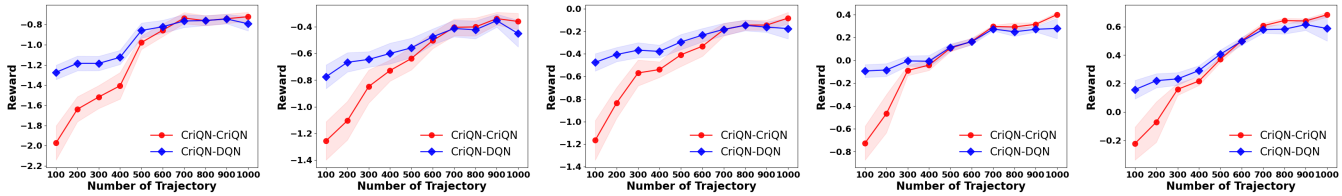


Fig. 6. Total Steps in Online Evaluation

2) Step Saving Comparison: Fig 6 shows the run-time steps for each critical policy. The X axis is the LongTR thresholds while the Y axis is the number of total steps from start to end point in the online evaluation. In Fig 6, before threshold 30%, all the four critical policies take similar steps. However, after 30%, CriQN-CriQN and CriQN-DQN take significant fewer steps than the DQN-CriQN and DQN-DQN. It suggests that the critical states identified by CriQN are more effective in reducing the number of steps in the trajectory. However, DQN-CriQN and DQN-DQN take more steps than the Full policy after 40%. Note that the goal of the RL-induced policy is to collect as much reward as possible, but not to find the shortcut path to the destination. So it suggests that inaccurate critical states can misguide the agent to take more steps to obtain reward. Overall, the results show that make optimal decisions on critical states could reduce the number of total decisions for achieving the goal.

3) Data-Efficiency for CriQN policy: From the online evaluation results in GridWorld, CriQN-CriQN and CriQN-DQN are better than DQN-CriQN and DQN-DQN and thus we could conclude that CriQN is better than DQN in identifying critical states, but there's no big difference between the two in selecting optimal actions. This is because both CriQN and DQN have enough data to induce an optimal policy and select the best action. However, what if we do not have enough data to train an optimal policy, how does the CriQN perform?

Fig 7 (a)-(e) show the GridWorld online performance of CriQN-CriQN vs. CriQN-DQN as the number of training trajectories increases. The X axis is the number of trajectories used to train the critical policies. The Y axis is the reward received by each critical policy. In this experiment, we applied different LongTR thresholds to identify critical states and the only difference is which RL policy makes the decisions on the critical states. The results show that when the training dataset is less than 500 trajectories, the CriQN-CriQN is



(a) T_{LongTR} : 10% (b) T_{LongTR} : 20% (c) T_{LongTR} : 30% (d) T_{LongTR} : 40% (e) T_{LongTR} : 50%

Fig. 7. CriQN vs. Original-DQN in Decision Making

worse than the CriQN-DQN across all five figures. When the training dataset is larger than 500 trajectories, they have similar performance. Our results demonstrate that CriQN-DQN can achieve the same or better performance than CriQN-CriQN with the same amount of data.

IV. Real-World Application: Sepsis Treatment

A. Two Medical Datasets

Sepsis is a life-threatening disease associated with a high mortality rate and costly medical treatment. Our datasets are the electronic health records (EHR) collected from two different medical systems: Christiana Care Health System (CCHS) in Newark, Delaware and Mayo Clinic in Rochester, Minnesota, USA.

1) CCHS Dataset: In total there are 210,289 visits and 9,029,493 events. By combining the International Classification of Diseases, Ninth Revision (ICD-9), and clinician rules, we sampled 1,800 positive septic shock trajectories and 1,800 negative trajectories (no shock), keeping the same distribution of age, gender, race, and the length of hospital stay. To impute the missing value, we applied the expert imputation rules that 1) the values of vital signs were carried forward for 8 hours, 2) the values of lab results were carried forward for 24 hours and, 3) the remaining missing values were imputed by mean values. The final dataset consists of 3,600 visits (50% shock, 50% no shock) and 84,160 events for which the average length of trajectories is 24 and the maximum length is 317.

2) Mayo Dataset: In total, there are 221,700 visits and 144,693,491 events. Similarly, by combining the ICD-9 and clinician rules, we sampled 2,205 positive septic shock trajectories and 2,205 negative trajectories (no shock), keeping the same distribution of age, gender, race, and length of hospital stay. To impute the missing value, we applied the same rule with the CCHS dataset. The final dataset includes 4,410 visits (50% shock, 50% no shock) and 392,850 events where the average trajectory length is 65 and the maximum trajectory length is 1160.

B. Experiment Setup

1) State, Action, Reward: Our states, actions and rewards were built based on the advice from clinicians with high domain expertise. *This definition and the following data preprocessing steps are determined by three leading clinicians with extensive experience on this subject.* The states are approximated from 15 sepsis-related clinical measurements, including 7 vital signs (HeartRate, PulseOx, Respiratory rate, Temperature, SystolicBP, DiastoicBP, Mean Arterial Pressure) and 8 lab results (Bands, BiliRubin, Blood Urea Nitrogen, FiO2, Creatinine, Lactate, Platelet, White Blood Cell).

Generally, medical treatments can be defined in both discrete and continuous action spaces; for example, a decision of whether a certain drug is administrated is discrete, while the dosage of drug is continuous. Continuous action space has been mainly handled with policy-based RL models such as actor-critic models [27], and it is generally only available for online RL. Since EHRs are offline data, and it is infeasible to search continuous treatment action by interacting with actual patients, we focus on discrete actions. Thus, we defined three discrete actions: two types of medical treatments (**antibiotic administration (A)** and **oxygen assistance (O)**) and **no treatment (N)**. The agent should learn a treatment policy that determines when and which types of treatments should be executed. More specifically, the two treatments can be applied simultaneously, which results in a total of four actions.

For the reward function, we defined four stages of sepsis, and the delayed rewards are set for each stage: infection ($\pm$5), inflammation ($\pm$10), organ failure ($\pm$20), and septic shock ($\pm$50). The designated negative reward was given when a patient enters into the corresponding stage, and its positive reward was given back when the patient recovers from the stage. As an example, if a patient recovers from the 'inflammation' stage, he receives a positive reward ($+10$) while if a patient enters the 'inflammation' stage, he gets a negative reward (-10). In this way, an optimal policy should keep patients from getting negative rewards and help them stay in non-negative states.

2) Offline Learning: The critical policy induction follows the same process as with the GridWorld III-B2, train InferNet model to infer immediate rewards, select the $T^{+}_{ShortTR}$ and $T^{-}_{ShortTR}$ thresholds, induce five CriDQN policies with different T_{LongTR} parameters from 10% to 50% and one DQN policy.

More specifically, we compared three types of critical policies: CriQN-CriQN, DQN-DQN and CriQN-DQN with two baseline policies: a fully-executed DQN (Full) policy and a Physician's policy. To train a policy that follows the physician actions, we followed the same procedure as described in [28] by using SARSA. Note that different from the GridWorld game, it is not ethical and undesirable to experiment the RL-induced policy on real patients. For the purpose of offline

evaluation, it is necessary to exclude part of the dataset from offline learning. Therefore, we conducted a 5-fold cross-validation and the dataset was split into 80% training and 20% test sets with the equal number of positive/negative shock trajectories.

3) Offline Evaluation: The effectiveness of critical policies were evaluated offline on the test dataset using two metrics: 1) septic shock rate and 2) percentage of nudges. In general, our expectation is that an effective critical policy will have the same low rate of septic shock as the Full and Physician policies, but with fewer nudges.

Septic Shock Rate: In similar fashion to prior studies ([28]–[30]), the induced policies were evaluated using the *septic shock rate*. The *septic shock rate* r_{shock} was first used in [15] and the assumption behind it is: when a septic shock prevention policy is indeed effective, the more the real treatments in a patient trajectory agree with the induced policy, the lower the chance the patient would get into septic shock; vice versa, the less the real treatments in a patient trajectory agree with the induced policy (more dissimilar), the higher the chance the patient would get into septic shock.

In this analysis, we compared three critical policies against the Full and Physician policies by looking at the septic shock rates for the 10% most similar group and the 10% least similar group (most dissimilar). To do so, first, for each trajectory, a similarity rate r_s between the policy's action and the actual physicians' action is calculated. The higher the r_s, the more similar the RL policy is to the physicians' treatment. Then we sort the trajectories by their similarity rate in ascending order and calculate the septic shock rate for the top 10% of trajectories with the highest similarity rate, referred as *10% most similar group* and the bottom 10% of trajectories with the least similarity rate, referred as *10% least similar group*. The septic shock rate is defined as: $r_{shock} = v_{shock}/v_s$, where v_s is the number of trajectories and v_{shock} is the number of positive-shock-trajectories. In general, we expect the septic shock rate for the 10% most similar group should be as low as possible, whereas those for the 10% least similar group to be higher.

Percentages of Decision States and Nudges: Percentage of decision States represents how often certain decisions are required to be made by the physician, while Percentage of Nudges indicates how often the induced policies would differ from physician's decisions and require the physician's attention. For critical policies, a nudge is needed if the state is critical and the physicians' action is different from the critical policy's action while for non-critical policy, a nudge is needed whenever the physicians' action differ from the policy. Note that when identifying critical states, the LongTR threshold functions as a hyper-parameter which approximately determines the percentage of the critical decisions in which the physicians must follow the corresponding policy (the higher the LongTR threshold, the more states will be considered as critical). In other words, it affects the percentage of critical states and nudges. In the offline evaluation, we explored the LongTR thresholds from 10% to 50% and stopped when either the critical policy beat the Full policy, or the threshold reaches 50%. After that, we investigated how fewer decisions and nudges are required to achieve the same effectiveness with the Full policy.

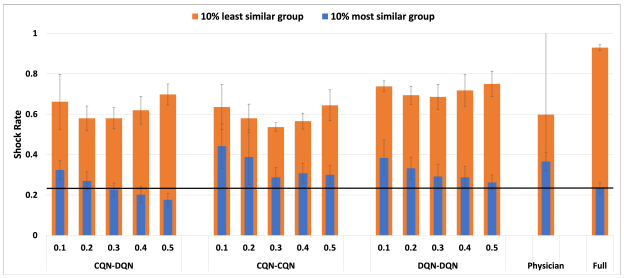


Fig. 8. Septic Shock Rate for CCHS

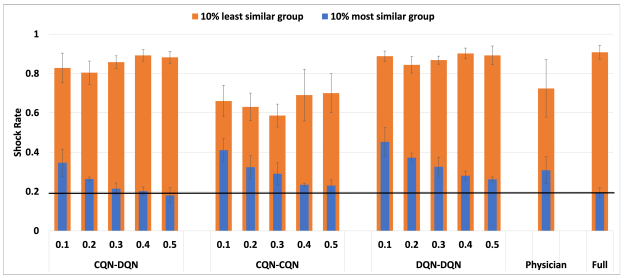


Fig. 9. Septic Shock Rate for Mayo

C. Results

1) Septic Shock Rate: Fig 8 and 9 shows the results of septic shock rate on CCHS and Mayo, respectively. In the X axis, there are two levels; the top level indicates the LongTR thresholds, and the second level shows the name of the policy. More specifically, for each bar, the wide column shows the septic shock rate of the 10% least group while the narrow column shows the 10% most similar group. For the critical policies, we only considered the similarity on the critical states. Note that the *horizontal black* line represents the septic shock rate of the Full policy in the 10% most similar group, which is our gold standard.

First, within each of the three types of critical policies on both datasets, there is a general trend that the larger the LongTR threshold, the lower the septic shock rate in the 10% most similar group. It suggests that as more states are considered critical and optimal actions are taken, the critical policy will become more and more effective in preventing septic shock. Furthermore, as expected, the septic shock rate in the 10% least group are significantly higher than the 10% most group. Such results suggest that our critical policies indeed have learned the common optimal treatments from the EHR dataset; following these treatments will reduce the odds of getting into septic shock, while not following them may greatly increase the likelihood of septic shock.

Second, when comparing the three types of critical policies, CriQN-DQN performs better than the other two across different LongTR thresholds on both datasets. It aligns with our GridWorld results that the best critical policy is CriQN-DQN which applying CriQN to identify critical states while original

DQN to select optimal actions. It is also important to note that the Physician policy fails to perform as well as the Full policy on both datasets and has a higher rate of septic shock for the 10% most similar groups of patients.

Finally, across the five T_{LongTR} thresholds, the CriQN-DQN with $T_{LongTR} = 0.5$ has the lowest sepsis rate for both CCHS and Mayo. When comparing the best critical policy setting with the corresponding Full policy (black horizontal bar), the CriQN-DQN with $T_{LongTR} = 0.5$ policy outperforms the corresponding Full policy on CCHS and the two policies are very close on Mayo. Thus, the CriQN-DQN with $T_{LongTR} = 0.5$ will be used for the following analysis for both CCHS and Mayo. Overall, our analysis shows that CriQN-DQN is the best critical policy and can be as effective as a fully-executed policy or even better.

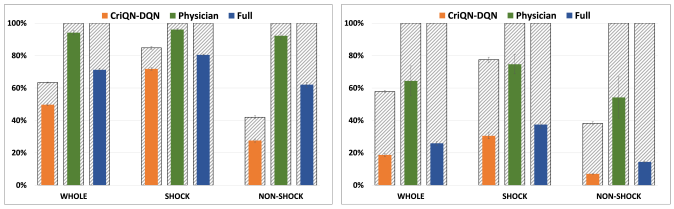


Fig. 10. Percentages of Decision States (transparent wide column) & nudges(solid narrow column) on different groups of patients, CCHS (Left) and Mayo (Right).

2) Percentage of Decision States and Nudges: Fig 10 illustrates the average percentages of decision states and nudges per trajectory identified by the corresponding policies in the *test dataset*. In Fig 10, we further split the patients into shock and non-shock groups to gain a better understanding of where the critical states and nudge happens.

First, the transparent wide columns show the percentage of decision states for each policy across all patient groups. Note that the CriQN-DQN policy uses LSTRs to identify "critical" states and it only requires decisions to be made in critical states, but for the Physician and Full policies, it requires decisions to be made in all states. Fig 10 shows that CriQN-DQN can identify around 60% of states as critical across both CCHS and Mayo, and specifically, about 80% of states are critical among shock patients, and only about 40% among non-shock patients.

Second, across all comparisons, the Physician policy has the most nudges. This suggests that physicians may not always follow a consistent treatment regimen. It is especially noticeable in CCHS since physician actions contradict its corresponding Physician Policy more than 90% of the time for both sepsis shock and non-shock patients; however, these differences are lower at Mayo: 75% for shock cases and 50% for non-shock patients.

Third, the proposed CriQN-DQN policy can significantly reduce the amount of decisions that physicians need to make through our HC-MA framework. Using the Full policy in our HC-MA framework would still require physicians to make *all* of decisions. Using CriQN-DQN requires them to make 60% of decisions. Decisions are saved especially evident in non-shock patients in that only about 40 percent of them are critical. Furthermore, physicians will receive fewer nudges. With CCHS, the CriQN-DQN is nudged 50% of times and less than 30% for non-shock patients, while at Mayo it is nudged less than 20% and less than 10% for non-shock patients. The majority of the nudge savings come from non-shock patients since it saves 60% of decisions requiring physician attention.

Finally, the number of non-shock patients (before sampling) is several times that of shock patients. It indicates that the critical policy could save tremendous amounts of nudges in real life. Moreover, in both medical systems, all the policies have more nudges on shock patients than that on the non-shock patients. It is reasonable that shock patients are experiencing more severe moments and require more attention. According to the results, CriQN-DQN requires the least amount of nudging, and this shrinkage is the result of reducing nudging on the non-shock patients while still paying close attention to the shock patients. This aligns with our expectations that the policy should alert on the critical moments and in the meantime, minimize the number of unnecessary alerts in order to avoid decision fatigue.

3) Case Study: To illustrate how the CriQN-DQN policy nudges physicians in the sepsis treatment, Fig 11 shows a case study on a shock patient and a non-shock patient from Mayo. In Fig 11, there are two levels in the X axis that the top level shows the event number in chronological order and the second level indicates the time after first arrival in the hospital. For the Y axis, there are two categories that 'Treatment' shows the real physicians' treatments and 'Nudge' indicates the suggested treatment from CriQN-DQN policy. More specifically, 'transparent' blocks in Nudge indicates either the decision is non-critical or the policy's treatment is the identical to that of the physician. In Fig 11, for the shock patient, our critical policy would nudge at the very start of the treatment process to bring attention to this patient and then continue to nudge the physician in order to suggest the best possible treatment, which indicates the patient is in a severe condition. In the meantime, for the non-shock patient, our critical policy only nudge the physician two times. Overall, the case study shows that our policy is capable of providing early warnings on shock patients and continuing to do so as the patient condition deteriorates. Furthermore, it would minimize the unnecessary alerts on the non-shock patients.

V. Related Work

Healthcare Workload and Alert: In healthcare systems, practitioners' workload is a critical concept affecting the quality of care and patient outcomes. The review by Fishbein et al. [31] identified objective measures of workload at four levels (task, patient, clinician, and unit). Workload measures at the task level included time to perform the task and task complexity; at the clinician level, measures included time spent on a task, number of procedures, and number of patients managed simultaneously; at the unit level, factors included work interruptions and unit activity. Clinician workload plays a critical role in clinician performance [32] and has been associated

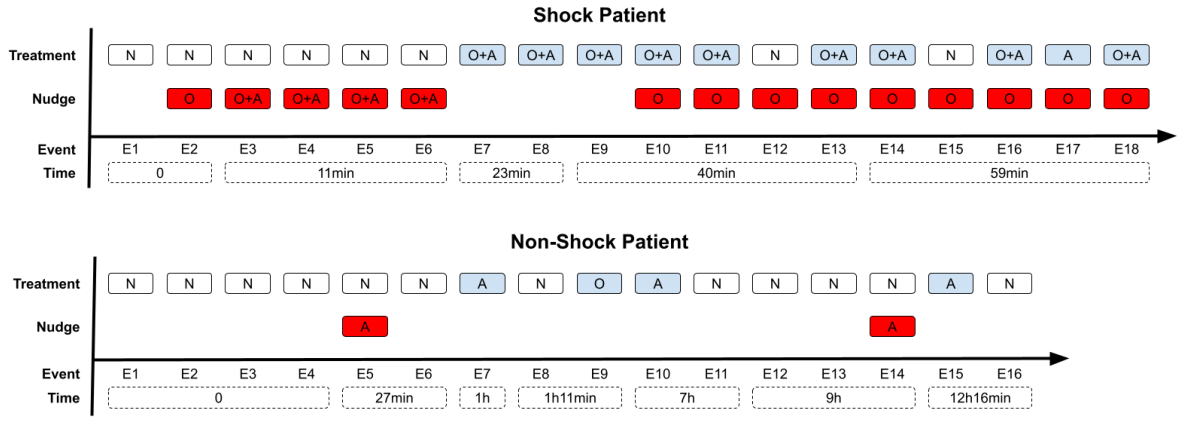


Fig. 11. Case Study: whole treatment trajectory to show the real physicians' treatments and Nudges from the critical policy.
For treatment: 'N' means no action, 'A' means antibiotic administration, 'O' means oxygen assistance, 'A+O' means apply two treatments together.

with provider burnout [33], [34], likelihood of medication error [34], mortality [35], [36], adverse events related to mechanical ventilation [37], length of stay (LOS) and procedure related infections [38]. Thus, even though it is important to improve treatment of patients, it is equally important to do so in a manner that does not increase the workload of clinicians. However, broad alerts can do little to improve performance while negatively impacting physician workflow. In one study, EHR alerts were automated for emergency physicians; these alerts were received frequently and most were perceived to not impact patient care (changing clinical management about 2% of the time) [39]. Clinical decision support systems that warn of irrelevant actions can result in alert fatigue which can actually lead providers to overlook important signals. Consequently, when implementing technology to improve care, it is important to consider provider alert fatigue by minimizing the number of unnecessary warnings.

Deep Reinforcement Learning: Recent advances in deep learning have allowed RL to work in complex interactive environments which was often impractical before. Recent work showed that RL can induce effective policies for a variety of tasks, such as game playing [26], [40], robotic control [41], [42], recommendation generation [43], [44] and also healthcare treatment [15] and [45]. However, all of the state-of-art RL algorithms focused on inducing effective policies. None of them considered interpreting, explaining and identifying critical decisions from RL induced policies.

LSTRs in Animal and Human Decision-making: The rise of computational neuroscience has allowed researchers to treat the brain as a super computing machine so as to understand the learning and decision-making process in animals and humans. A lot of studies have shown that many key RL signals exist in the human and animal brains during the learning and decision-making process. In animal studies, Morris [46] and Roesch [47] trained monkeys and rats to perform a binary choice task with different actions associated with different sizes of rewards. They found that monkeys and rats maintain Q-values in their brain and prefer to choose the action with higher Q-value. In Sul's work [48], he trained rats in a maze to choose to go left or go right to get rewards. He found that some brain neurons encode the Q-value difference signal when making decisions and this Q-value difference reflects the desirability of choosing an action. In human studies, Li [49] also found the Q-value signals in the human brain when performing a two-armed bandit task. In Samuel's experiment [20], human participants took a series of binary choices between (small, early) and (big, later) monetary rewards. The results showed that there are two separate systems in our brain to deal with immediate rewards and delayed rewards. In summary, prior research has shown that Q-values and rewards are widely used in animal and human decision-making and RL is one of the most promising frameworks to model the decision-making process in humans.

VI. Conclusions

In this study, we explored a Critical-DRL approach to identify critical decisions in both a synthetic simple GridWorld game and a real-world healthcare dataset. Our results from both tasks showed that the CriQN is significantly better than the original DQN in identifying critical states. For GridWorld, the performance of CriQN-CriQN and CriQN-DQN are very close and both perform better than DQN-CriQN and DQN-DQN while for sepsis prevention, CriQN-DQN performs the best. Overall, the best critical policy is using CriQN to identify critical states but utilizing original DQN to select optimal actions. Our results on sepsis treatment show that the induced critical policy could reduce the percentage of nudges while keeping the septic shock rate as low as a fully-executed policy. In summary, this paper provides some evidence for employing our proposed general human-controlled machine-assisted (HC-MA) DRL framework in healthcare domain where physicians are always overloaded and efficient alert is needed.

REFERENCES

[1] M. W. Stanton, *Hospital nurse staffing and quality of care*. Agency for Healthcare Research and Quality Rockville, MD, 2004.

[2] J. S. Weissman, J. M. Rothschild, E. Bendavid, P. Sprivulis, E. F. Cook, R. S. Evans, Y. Kaganova, M. Bender, J. David-Kasdan, P. Haug *et al.*, "Hospital workload and adverse events," *Medical care*, vol. 45, no. 5, pp. 448–455, 2007.

[3] P. Carayon and A. P. Gurses, "Nursing workload and patient safety—a human factors engineering perspective," 2008.

[4] M. M. Cohen, L. L. O'Brien-Pallas, C. Copplestone, R. Wall, J. Porter, and D. K. Rose, "Nursing workload associated with adverse events in the postanesthesia care unit." *Anesthesiology*, vol. 91, no. 6, pp. 1882–1890, 1999.

[5] F. Al-Kandari and D. Thomas, "Perceived adverse patient outcomes correlated to nurses' workload in medical and surgical wards of selected hospitals in kuwait," *Journal of Clinical Nursing*, vol. 18, no. 4, pp. 581–590, 2009.

[6] S. Hugonnet, J. Chervrolet, and D. Pttet, "The effect of workload on infection risk in critically ill patients," *Crit Care Med*, vol. 35, no. 1, pp. 76–81, 2007.

[7] B. Mitchell, A. Gardner, P. Stone, L. Hall, and M. Pogorzelska-Maziara, "Hospital Staffing and Health Care - Associated Infections: A Systematic Review of the Literature," *The Joint Commission Journal on Quality and Patient Safety*, vol. 44, no. 6, pp. 613–622, 2018.

[8] B. L. Hooey, D. B. Kaber, J. A. Adams, T. W. Fong, and B. F. Gore, "The underpinnings of workload in unmanned vehicle systems," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 5, pp. 452–467, 2017.

[9] P. A. Swiger, D. E. Vance, and P. A. Patrician, "Nursing workload in the acute-care setting: A concept analysis of nursing workload," *Nursing Outlook*, vol. 1, no. 2, pp. 131–43, 2011.

[10] P. Hoonakker, P. Carayon, A. G. R. Brown, K. McGuire, A. Khunlertkit, and J. M. Walker, "Measuring workload of icu nurses with a questionnaire survey: the nasa task load index (tlx)," *IIE Trans Healthcare Syst Eng*, vol. 64, pp. 244–254, 2016.

[11] E. H. Ofstad, J. C. Frich, E. Schei *et al.*, "Clinical decisions presented to patients in hospital encounters: a cross-sectional study using a novel taxonomy," *BMJ Open*, vol. 8, 2018. [Online]. Available: https://bmjopen.bmj.com/content/bmjopen/8/1/e018042.full.pdf

[12] G. A. Pignatiello, R. J. Martin, and R. L. H. Jr, "Decision fatigue: A conceptual analysis," *Journal of Health Psychology*, vol. 25, no. 1, pp. 123–135, 2020.

[13] J. L. Allan, D. W. Johnston, D. J. H. Powell, B. Farquharson, M. C. Jones, G. Leckie, and M. Johnston, "Clinical decisions and time since rest break: An analysis of decision fatigue in nurses," *Health Psychology*, vol. 38, no. 4, pp. 318–324, 2019.

[14] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 2018.

[15] A. Raghu, M. Komorowski, I. Ahmed, L. Celi, P. Szolovits, and M. Ghassemi, "Deep Reinforcement Learning for Sepsis Treatment," *arXiv preprint arXiv:1711.09602*, 2017.

[16] S. Y. E. B. Hongseok Namkoong, Ramtin Keramati, "Off-policy policy evaluation for sequential decisions under unobserved confounding," *arXiv preprint arXiv:2003.05623*, 2020.

[17] A. D. F. G. J. A. C. M. Aurelie Bourgoin, Marc Leone, "Increasing mean arterial pressure in patients with septic shock: effects on oxygen variables and renal function," *Crit Care Med*, vol. 33, no. 4, pp. 780–786, 2005.

[18] A. Rhodes, L. Evans, W. Alhazzani *et al.*, "Surviving sepsis campaign: International guidelines for management of sepsis and septic shock: 2016." *Intensive Care Med*, vol. 43, pp. 304–377, 2017. [Online]. Available: https://doi.org/10.1007/s00134-017-4683-6

[19] V. Liu, V. Fielding-Singh, J. Greene *et al.*, "The timing of early antibiotics and hospital mortality in sepsis," *AJCMED*, vol. 196, no. 7, pp. 856–863, 2017.

[20] S. M. McClure, D. I. Laibson, G. Loewenstein, and J. D. Cohen, "Separate neural systems value immediate and delayed monetary rewards," *Science*, pp. 503–507, 2004.

[21] L. Torrey and M. E. Taylor, "Teaching on a budget: Agents advising agents in reinforcement learning," *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13*, pp. 1053–1060, 2013.

[22] M. Zimmer, P. Viappiani, and P. Weng, "Teacher-student framework: A reinforcement learning approach," *AAMAS Workshop Autonomous Robots and Multirobot Systems*, 2013.

[23] O. Amir, E. Kamar, A. Kolobov, and B. J. Grosz, "Interactive teaching strategies for agent training," *the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, pp. 804–811, 2016.

[24] A. Fachantidis, M. E. Taylor, and I. P. Vlahavas, "Learning to teach reinforcement learning agents," *Machine Learning and Knowledge Extraction*, 2017.

[25] M. S. Ausin, H. Azizsoltani, S. Ju, Y. J. Kim, , and M. Chi, "Infernet for delayed reinforcement tasks: Addressing the temporal credit assignment problem," *CoRR abs/2105.00568*, 2021.

[26] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, and et al., "Human-level control through deep reinforcement learning." *Nature*, no. 518, pp. 529–533, 2015.

[27] T. Lillicrap, J. Hunt, A. Pritzel *et al.*, "Continuous control with deep reinforcement learning," in *ICLR*, 2016.

[28] H. Azizsoltani, Y. J. Kim, M. S. Ausin, T. Barnes, and M. Chi, "Unobserved is not equal to non-existent: Using gaussian processes to infer immediate rewards across contexts," *IJCAI*, pp. 1974–1980, 2019.

[29] M. Komorowski, L. Celi, O. Badawi *et al.*, "The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care," *Nat Med*, vol. 24, 2018. [Online]. Available: https://doi.org/10.1038/s41591-018-0213-5

[30] A. Raghu, M. Komorowski, I. Ahmed, L. A. Celi, P. Szolovits, and M. Ghassemi, "Deep reinforcement learning for sepsis treatment," 2017.

[31] D. Fishbein, S. Nambiar, K. McKenzie, M. Mayorga, K. Miller, K. Tran, L. Schubel, J. Agor, T. Kim, and M. Capan, "Objective measures of workload in healthcare: a narrative review," *International Journal of Health Care Quality Assurance*, vol. 33, pp. 1–17, 2019.

[32] L. M. Mazur, P. R. Mosaly, C. Moore, E. Comitz, F. Yu, A. D. Falchook, M. J. Eblan, L. M. Hoyle, G. Tracton, B. S. Chera, and L. B. Marks, "Toward a better understanding of task demands, workload, and performance during physician-computer interactions," *Journal of the American Medical Informatics Association*, vol. 23, pp. 1113–1120, 2016.

[33] I. Portoghese, M. Galletta, R. C. Coppola, G. Finco, and M. Campagna, "Burnout and workload among health care workers: The moderating role of job control," *Safety Health and Work*, vol. 5, no. 3, pp. 152–157, 2014.

[34] R. J. Holden, M. C. Scanlon, N. R. Patel, R. Kaushal, K. H. Escoto, R. L. Brown, S. J. Alper, J. M. Arnold, T. M. Shalaby, K. Murkowski, , and B.-T. Karsh, "A human factors framework and study of the effect of nursing workload on patient safety and employee quality of working life," *BMJ quality safety*, vol. 20, no. 1, pp. 15–24, 2011.

[35] C. P. S. P. F. A. F. D. J.-J. L. V. P. J. N. T. R. A.-M. S. A. D. Antoine Neuraz, Claude Guérin, "Patient mortality is associated with staff resources and workload in the icu: A multicenter observational study," *Crit Care Med*, vol. 43, no. 8, pp. 1587–1594, 2015.

[36] L. H. A. W. S. D. M. S. A. M. R. R. L. C. T. P. G. R. C. Jane E.Ball, Luk Bruyneel, "Post-operative mortality, missed care and nurse staffing in nine countries: A cross-sectional study," *International Journal of Nursing Studies*, vol. 78, pp. 10–15, 2018.

[37] J. M. A. L. Z. C. L. V. M. F. S. A. M. d. S. Fernando Lamy Filho, Antonio A. M. da Silva, "Staff workload and adverse events during mechanical ventilation in neonatal intensive care units," *J Pediatr (Rio J)*, vol. 87, no. 6, pp. 487–492, 2011.

[38] C. d. O. R. T. M. A. d. S. B. G. M. S. S. d. M. Ana Maria Müller de Magalhaes, Diovane Ghignatti da Costa, "Association between workload of the nursing staff and patient safety outcomes," *Revista da Escola de Enfermagemda USP*, no. 51, 2017.

[39] L. N. L. M. M. M. R. O. Brett Todd, Nashid Shinthia, "Impact of electronic medical record alerts on emergency physician workflow and medical management," *The Journal of Emergency Medicine*, vol. 60, no. 3, pp. 390–395, 2021.

[40] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning." *In NIPS Deep Learning Workshop*, 2013.

[41] S. Levine, C. Finn, T. Darrell, and P. Abbeel., "End-to-end training of deep visuomotor policies." *JMLR*, no. 17(39), pp. 1–40, 2016.

[42] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi., "Target-driven visual navigation in indoor scenes using deep reinforcement learning." *ICRA*, 2017.

[43] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N. J. Yuan, X. Xie, and Z. Li., "Drn: A deep reinforcement learning framework for news recommendation." *World Wide Web Conference*, pp. 167–176, 2018.

[44] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, and J. Tang., "Deep reinforcement learning for page-wise recommendations." *In Proceedings of the 12th ACM Conference on Recommender Systems.*, pp. 95–103, 2018.

[45] D. W. O. G. M. K. L.-w. H. L. A. R. A. F. F. D.-V. Xuefeng Peng, Yi Ding, "Improving sepsis treatment strategies by combining deep and kernel-based reinforcement learning," *arXiv preprint arXiv:1901.04670*, 2019.

[46] G. Morris, A. Nevet, D. Arkadir, E. Vaadia, and H. Bergman, "Midbrain dopamine neurons encode decisions for future action," *Nature Neuroscience*, vol. 9, no. 8, pp. 1057–1063, 2006.

[47] M. R. Roesch, D. J. Calu, and G. Schoenbaum, "Dopamine neurons encode the better option in rats deciding between different delayed or sized rewards," *Nature Neuroscience*, vol. 10, no. 12, pp. 1615–1624, 2007.

[48] J. H. Sul, S. Jo, D. Lee, and M. W. Jung, "Role of rodent secondary motor cortex in value-based action selection," *Nature Neuroscience*, vol. 14, no. 9, pp. 1202–1208, 2011.

[49] J. Li and N. D. Daw, *Signals in Human Striatum Are Appropriate for Policy Update Rather than Value Prediction*, 2011, vol. 31, no. 14.