Proceedings of Machine Learning Research 129:433–448, 2020 ACML 2020

Inferring Continuous Treatment Doses from Historical Data via Model-Based Entropy-Regularized Reinforcement Learning

Jianxun Wang JWANG75@NCSU.EDU
David Roberts DLROBER4@NCSU.EDU
North Carolina State University

Andinet Enquobahrie ANDINET.ENQU@KITWARE.COM
Kitware, Inc.

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

Developments in Reinforcement Learning and the availability of healthcare data sources such as Electronic Health Records (EHR) provide an opportunity to derive data-driven treatment dose recommendations for patients and improve clinical outcomes. Recent studies have focused on deriving discretized dosages using offline historical data extracted from EHR. In this paper, we propose an Actor-Critic framework to infer continuous dosage for treatment recommendation and demonstrate its advantage in numerical stability as well as interpretability. In addition, we incorporate a Bayesian Neural Network as a simulation model and probability-based regularization techniques to alleviate the distribution shift in off-line learning environments to increase practical safety. Experiments on a real-world EHR data set, MIMIC-III, show that our approach can achieve improved performance while maintaining similarity to expert clinician treatments in comparison to other baseline methods.

1. Introduction

Recent developments in Reinforcement Learning (RL) have inspired its applications in complex and high-risk domains, such as patient treatment recommendations in Intensive Care Units (ICU). A particularly interesting treatment recommendation problem is the treatment dose prescribed to the patient. RL algorithms model the treatment process as a temporal sequence of treatment and observation events. Provided with a reward function based on patients' survival or response to treatment, these techniques are capable of learning a treatment strategy that recommends the dose of treatment for patients at each treatment interval to maximize patient outcomes. Recent studies (*e.g.* Komorowski et al. (2018); Raghu et al. (2017); Peng et al. (2018)) using historical data such as Electronic Health Records (EHR) have demonstrated such potential. However, deriving an interpretable and practically-safe treatment strategy using only historical data remains a challenging task.

An important yet overlooked design aspect of RL for treatment recommendation is the encoding of treatment options. It is crucial for RL algorithms to derive meaningful strategies that enable clinicians to interpret intermediate suggestions. Existing methods (*e.g.* Komorowski et al. (2018); Raghu et al. (2017)) discretize the dose based on quantile or a fixed range and consider discrete volume as categorical options. While modeling discrete doses has its benefits such as balanced data distribution or tractable search space, it can introduce bias in the interpretation of model output or even produce questionable results. In addition, since RL algorithms model the treatment as

categorical data, the ordinal relationship between different doses can be lost during the training process. Deriving continuous treatment doses is a desirable alternative to avoid these problems.

Limiting to historical data for training increases difficulty in deriving effective treatment strategies, though modeling continuous doses may amplify the challenge because the search space of treatments may become intractable. Due to ethical concerns and technical difficulties, it is not feasible to train RL models for treatment recommendation by using model-recommended treatments in real-world environments where policy exploration can result in adverse health consequences. Such restrictions requires RL models to learn off-line instead of in an online, trial-and-error fashion. Past studies suggest that RL models may capture inaccurate dynamics of patients' state transitions due to insufficient exploration and produce ineffective suggestions (Levine et al. (2020)). This may also cause models to incorrectly identify dangerous treatments as optimal or overlook optimal options (Gottesman et al. (2018)), leading to severe consequences in the real-world.

There are generally three data-driven approaches to alleviate this problem. Imitation learning aims to directly mimic clinician treatments in historical data through methods such as Supervised Learning or RL based on a reward function retrieved by Inverse Reinforcement Learning (IRL) (Abbeel and Ng (2004)). Its downside is the algorithm cannot identify whether the treatment will lead to patients' long-term survival. Another approach applies regularization to RL based on different criteria, for example policy entropy (Ziebart (2010)) or difference to the behavior in historical data (Kumar et al. (2019)). However, insufficient exploration may still prevent such approach from discovering an effective strategy. Finally, model-based RL learns an environment model to facilitate exploration (Kaiser et al. (2019)) but RL models may adopt errors in the environment model for a partially-observable environment like patient's physiology system. These approaches each has drawbacks but possesses unique strengths in tackling the challenge of off-line learning.

In this paper, we propose a Model-based Regularized Actor-Critic framework that combines the aforementioned methods to derive effective continuous treatment doses using historical data only. Instead of directly producing treatment doses, we design the model to output its distribution for interpretation. To achieve this, we adopt a probabilistic inference approach of RL using an uncertainty component for regularization. In addition, we incorporate a Bayesian Neural Network (Wen et al. (2018)) to simulate the partially-observable environment. It allows us to explore the environment and deploy a sample-based Deep Inverse Reinforcement Learning framework called GAN-GCL (Fu et al. (2017)). Rather than directly use the derived reward from GAN-GCL, we include it in the augmented regularization to achieve automatic balance between heuristically designed reward and data driven reward. Our experiments on the MIMIC-III dataset (Johnson et al. (2016)) show that our approach improves model performance over other baseline models while maintaining similarity to expert demonstrations.

2. Related Work

Previous studies deploying RL in treatment recommendation using EHR focused on categorical or discretized treatment and heuristically-designed reward functions based on different medical objectives in treatment procedures. Komorowski et al. (2018) modeled discrete sepsis treatment doses and with a discrete state space using Value Iteration and a terminal reward based on patients' 90-day survival. Raghu et al. (2017) extended the approach to continuous state spaces using Double-Deep Q-Networks and augmented the reward function based on medical knowledge. Prasad et al. (2017) utilized Q-learning with a neural network as a function approximator to derive discretized sedation

doses and binary ventilator support. However, several studies (Gottesman et al. (2018); Mihatsch and Neuneier (2002)) suggest that these systems may recommend dangerous treatments because of off-line learning using historical data only and a lack of expert guidance.

Additional studies utilized different techniques to alleviate such problem. Yu et al. (2019) imitated clinician treatments in a RL paradigm using a reward function derived by IRL. Wang et al. (2018) introduced supervised cross-entropy loss in the policy update to incorporate expert guidance from historical data and reduce risky treatment suggestions. Peng et al. (2018) achieved similar effect by combining kernel methods and Deep RL. Raghu et al. (2018) deployed a neural network as an environment model to predict patients' state changes and to facilitate environment exploration.

In this paper, we introduce a framework that combines the benefit of expert guidance and model-based reinforcement learning. Our work is built on Soft Actor Critic (SAC) framework (Haarnoja et al. (2018)) that allows us to model a continuous probability distribution of treatment doses by including an Actor-Critic framework and entropy regularization. We augment the entropy regularization in SAC to incorporate a probabilistic guidance model learned from expert historial data. A simple expert model can be a supervised learning model that directly minimizes the different between model output and historical data. However, an IRL model can be more effective. We include a Bayesian Neural Network (Wen et al. (2018)) as a simulator for the partially-observable environment to deploy a sample-based Deep IRL model called GAN-GCL (Fu et al. (2017)) and enable environment exploration. We use the RL model learned from the GAN-GCL-derived reward as the expert model for regularization and train the base model using pre-defined reward.

3. Background

We formulate the patient treatment procedure as a sequential trajectory $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, ...)$. It can be formally described by a Markov Decision Process (MDP) defined as a tuple (S, A, T, r). At each time step t, the clinician perceives patient's state $s_t \in S$ and selects doses for each modeled treatment option to form an action $a_t \in [0, 1]^k$. To account for the maximum dose and and regular volume of different treatments, we consider action as a k-dimension vector and scale each treatment dose into $[0, 1]$. Given the state-action pair (s_t, a_t), the clinician receives a feedback or reward $r : S \times A \to \mathbb{R}$. Patient's transition into subsequent state s_{t+1} is implicitly captured by transition probability density $T : S \times A \times A \to [0, \infty]$. The objective of the RL algorithm is to derive an optimal policy $\pi^*(a_t|s_t)$, denoting the conditional probability distribution of a_t, such that following it will maximize the expected sum of reward $\sum_t \mathbb{E}[\gamma^t r(s_t, a_t)]$, where $\gamma \in [0, 1]$ is a discount factor.

3.1. Regularized Reinforcement Learning Model

Recent studies (Levine (2018)) suggest that optimizing toward the original objective function alone may be insufficient in partially-observable environments. A common augmentation is adding a regularization term to the original objective. While there are many different forms of regularization, the objective and the result policy can be summarized as:

$$\pi^* = \arg\max_{\pi} \sum_t \mathbb{E}[\gamma^t (r(s_t, a_t) + \alpha \mathcal{R}(s_t))]. \tag{1}$$

α is the temperature parameter that balances the relative importance between the reward r and the regularization term $\mathcal{R}$. When $\alpha = 0$, the objective function reduces to the original one. Two types of regularization are of interest. The first one, called Maximum Entropy Reinforcement Learning

(Ziebart et al. (2008); Haarnoja et al. (2018)), uses the entropy of the policy $\mathcal{H}(\pi(\cdot|s_t))$ to equivalently minimize the KL-divergence between the trajectory distributions generated by the current policy π and the unknown optimal policy π^*. The second one, for example BEAR (Kumar et al. (2019)), aims to minimize the difference between the current policy and a guidance such as treatments in the historical data. BEAR used a sample-based approach and achieved this by minimizing the maximum mean discrepancy between actor policy and unknown guidance. Our approach is based on KL-divergence and utilizes a learned guidance policy that generalizes the expert guidance.

To model the probability distribution of continuous treatment doses, we include an Actor-Critic framework that maintains a parameterized policy $\pi_\phi(a_t|s_t)$ ("actor") and a parameterized Q-value function $Q_\theta(s_t, a_t)$ ("critic") to evaluate the actor using the regularized objective. The training process alternates between policy evaluation and policy improvement. In policy evaluation, we update the Q-value function or its approximator:

$$Q_\theta(s_t, a_t) = r(s_t, a_t) + \gamma \mathbb{E}_{s_{t+1}, a_{t+1} \sim \pi(\cdot|s_{t+1})}[Q_\theta(s_{t+1}, a_{t+1}) + \alpha \mathcal{R}(s_{t+1})] \tag{2}$$

In policy improvement, the policy is selected or updated according to the regularized Q-value:

$$\pi_{new} = \arg\max_\pi \mathbb{E}_{s_t \sim D, a_t \sim \pi(\cdot|s_t)}[Q(s_t, a_t) + \alpha \mathcal{R}(s_t)]. \tag{3}$$

The temperature parameter α can be treated as a fixed hyperparameter or updated according to the re-written dual objective (Haarnoja et al. (2018)) to provide automatic balance between regularization and accumulated reward:

$$\max_\pi \mathbb{E}[\sum_t r(s_t, a_t)] \text{ s.t. } \mathbb{E}[\mathcal{R}(s_t)] \geq \mathcal{K}. \tag{4}$$

3.2. Inverse Reinforcement Learning and GAN-GCL

RL algorithms require an explicit reward function as the optimization goal. Inverse Reinforcement Learning (IRL) (Abbeel and Ng (2004); Boularias et al. (2011)) was proposed to derive a data-driven reward function from expert demonstrations based on the assumption that expert's actions are suboptimal or near-optimal in an attempt to maximize an unknown reward function. GAN-GCL (Fu et al. (2017)) formulates IRL through generative adversarial modeling. In GAN-GCL, a policy model interacts with the real-world or simulation environment to generate trajectories and a discriminator is trained to distinguish generated trajectories from expert demonstrations. It assumes the discriminator takes the form (with $\pi(a|s)$ precomputed):

$$D_\psi(s, a, \pi(a|s)) = \frac{\exp(\hat{r}_\psi(s, a))}{\exp(\hat{r}_\psi(s, a)) + \pi(a|s)}. \tag{5}$$

As a result, discriminator's output can be interpreted as the probability of a (s_t, a_t) pair coming from expert demonstration. The discriminator can be updated using binary cross entropy loss.

GAN-GCL will then update the policy model with reward function $f_\psi(s_t, a_t)$ using any type of policy optimization for RL. Summation of this reward function over the entire trajectory space is equivalent to the entropy regularized objective:

$$\begin{aligned} f_\psi(s_t, a_t) =& \log(D_\psi(s_t, a_t)) - \log(1 - D_\psi(s_t, a_t)) \\ =& \hat{r}_\psi(s_t, a_t) - \log \pi(a_t|s_t). \end{aligned} \tag{6}$$

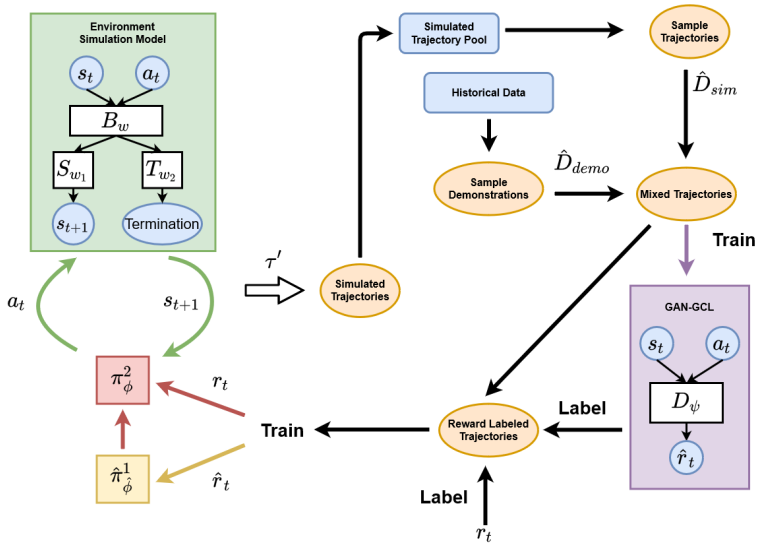

Figure 1: Model-Based Regularized Actor Critic Architecture. $\hat{\pi}^1_{\hat{\phi}}$ is the guidance policy and π^2_ϕ is the RAC policy. Only actors are presented here while critics are omitted for clarity.

4. Methodology

In this section, we present our proposed Model-Based Regularized Actor Critic framework to predict continuous treatment doses. The overall architecture is summarized in Figure 1.

4.1. Continuous Treatment Dose Modeling

We scale the dose of different treatment options into $[0, 1]$ based on their maximum volume. The action space is a k-dimension vector where k is the number of treatment options. To adapt to multi-dimensional continuous treatment doses in $[0, 1]^k$, we model the policy as an independent, truncated, normal distribution with parameters (μ_t, σ_t) that is conditionally dependent on the patient's state. We expect the underlying modeling of patients' states should explain all correlation among modeled treatments. Therefore, it is safe to assume conditional independence given a patient's state. We will explain more in detail about how we generate the distribution parameters later.

Using a truncated distribution allows actions on the closed borders, but it can lead to elevated probability density and cause exploding gradients during training. To avoid this, we clip the mean of each dimension according to the corresponding variance to ensure at least 5% of the cumulative density remains within the range. b_{lower} and b_{upper} are the boundaries of the truncated normal distribution which take values of 0 and 1 respectively:

$$\mu_t = \max\{b_{lower} - z_{.95}\sigma_t, \min\{\mu_t, b_{upper} + z_{.95}\sigma_t\}\} \tag{7}$$
$$P(x \leq z_{.95}) = 0.95, x \sim \mathcal{N}(0, 1).$$

4.2. Environment Modeling

We simulate patients' state transitions using a pre-trained Bayesian Neural Network (BNN) to account for the partially-observable environment. We design the BNN as a multitask model that pre-

dicts state value changes and trajectory termination for the next time step. The environment model uses a shared Bayesian model B_w with Flipout estimator (Wen et al. (2018)) to extract shared features and produce noise. The state change estimation S_{w_1} and trajectory termination prediction T_{w_2} are connected to B_w. These two models contain fully-connected layers only since the environmental noise should be accounted for in both tasks. B_w, S_{w_1}, and T_{w_2} are combined together to form the environment model *Env* and are updated by simultaneously minimizing three loss functions: MSE loss from state change prediction Δs_t, categorical cross entropy loss from trajectory termination prediction ϱ_t, and KL-Divergence loss as the regularizer from the Bayesian layer weights' posterior distribution $q(w|\theta_w)$ to the prior standard normal distribution $p(w)$:

$$J_1(w, w_1) = \mathbb{E}[(\Delta s_t - \hat{\Delta s_t})^2] \tag{8}$$

$$J_2(w, w_2) = \mathbb{E}[-\sum P(\varrho_t) \log P(\hat{\varrho_t})] \tag{9}$$

$$J_3(w) = D_{KL}(q(w|\theta_w)||p(w)). \tag{10}$$

An actor policy will interact with *Env* to generate the simulated trajectory. We first sample a seed state from the initial states in historical data. The seed state will be sent to the actor policy to produce the action for the initial seed state. *Env* receives the initial state and the produced action to predict the state change Δs_t and trajectory termination. If the trajectory does not terminate, Δs_t will be added to s_t to compute s_{t+1}. Such interaction will continue until the maximum trajectory length t_{max} is met.

4.3. Regularized Actor Critic

As mentioned above, we incorporate an Actor-Critic framework to model the distribution of continuous treatment doses. To encode continuous state and action values, we use a deep neural network as a function approximator for both the critic and the actor. The neural network in our study has two hidden layers and the tanh activation function for hidden layers. The critic $Q_\theta(s_t, a_t)$ receives the continuous value of patient state and corresponding action to produce an estimation of the regularized objective as the Q-function. The actor $\pi_\phi(a_t|s_t)$ receives patient state and produces the parameters of the policy distribution as output, in this case (μ_t, σ_t).

To update the parameters of the function approximators for the critic and the actor, we modify the objective functions for $Q_\theta(s_t, a_t)$ and $\pi_\phi(a_t|s_t)$ in Equation 2 and 3 into loss functions that can be updated using gradient methods. For the critic $Q_\theta(s_t, a_t)$, we use MSE loss to directly minimize the difference between the current estimation Q_θ and target value $\hat{y}$. We maintain a target Q network $Q_{\bar{\theta}}$ with exponential moving average of Q_θ's weights (Van Hasselt et al. (2016)) to stabilize the update of Q-values.

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim D}[\frac{1}{2}(Q_\theta(s_t, a_t) - \hat{y}_t)^2] \tag{11}$$

$$\hat{y}_t = r(s_t, a_t) + \gamma \mathbb{E}_{a_{t+1}\sim\pi(\cdot|s_{t+1})}[Q_{\bar{\theta}}(s_{t+1}, a_{t+1}) - \alpha\mathcal{R}(s_{t+1})]. \tag{12}$$

The function approximator of $Q_\theta(s_t, a_t)$ can be updated using semi-gradient descent with the gradient computed for estimation only (Sutton and Barto (2018)).

$$\nabla_\theta J_Q(\theta) = (Q_\theta(s_t, a_t) - \hat{y}_t)\nabla_\theta Q_\theta(s_t, a_t). \tag{13}$$

For the actor $\pi_\phi(a_t|s_t)$, we use the new Q-value to guide its update. Because the action is sampled from the policy, the gradient will backpropagate through the estimated Q-value to update the actor's parameters. If α equals 0, such a loss function is equivalent to DDPG (Lillicrap et al. (2015)), which is a deterministic actor critic algorithm.

$$J_\pi(\phi) = \mathbb{E}_{s_t \sim D, a_t \sim \pi_\phi}[-Q_\theta(s_t, a_t) + \alpha \mathcal{R}(s_t)]. \tag{14}$$

In practice, the policy function approximator produces the parameters that define the conditional distribution of a. Action sampling from the policy required in $J_\pi(\phi)$ can be achieved by a reparameterization trick. Hence, we apply the following gradient to minimize $J_\pi(\phi)$, where $f_\phi(\epsilon_t, s_t) = (\mu_t, \sigma_t)$ is the function to reparameterize noise ϵ_t sampled from $\mathcal{N}(0, 1)$ with policy parameters:

$$\nabla_\phi J_\pi(\phi) = \nabla_\phi \alpha \mathcal{R}(s_t) + \nabla_{a_t}(-Q_\theta(s_t, a_t) + \alpha \mathcal{R}(s_t)) \nabla_\phi f_\phi(\epsilon_t, s_t). \tag{15}$$

During the model training process, if a dynamic α parameter is needed, we transform the dual optimization problem in Equation 4 into the Lagrangian

$$\min_\alpha \max_\pi \mathbb{E}_{a_t \sim \pi}[r(s_t, a_t) + \alpha \mathcal{R}(s_t) - \alpha \mathcal{K}]. \tag{16}$$

Based on the previous equation, the temperature parameter is optimized through dual gradient descent (Boyd and Vandenberghe (2004)) with loss function

$$J(\alpha) = \mathbb{E}_{a_t \sim \pi}[\alpha \mathcal{R}(s_t) - \alpha \mathcal{K}]. \tag{17}$$

We consider two types of regularization based on probability theory. The first is the entropy of the actor policy; otherwise, we observed that the variance of distribution would never be updated. In this case,

$$\mathcal{R}^1(s_t) = -\log(\pi(a_t|s_t)), a_t \sim \pi(\cdot|s_t) \tag{18}$$

To include expert knowledge, we consider a guidance policy $\hat{\pi}$ and we augment the actor policy's entropy regularization with cross entropy of the guidance policy relative to the actor policy. The resulting regularization is the KL-divergence from the guidance to the actor:

$$\mathcal{R}^2(s_t) = -\log(\pi(a_t|s_t)) + \log(\hat{\pi}(a_t|s_t)), a_t \sim \pi(\cdot|s_t) \tag{19}$$

The guidance policy $\hat{\pi}$ is another trained model with the same model architecture as the actor. It generalizes the expert knowledge in historical data. Combining with KL-divergence, it also provides an opportunity to combine benefits from different signal sources or learning algorithms. A straightforward approach is to use a supervised model trained using negative log likelihood loss as guidance. We examine the effect of using the policy model learned based on the derived reward function from GAN-GCL. Under this formulation, the heuristically-designed reward and GAN-GCL derived reward are indirectly combined together. The relative importance between these two are balanced by the temperature parameter α that is either fixed or automatically adjusted.

We update the guidance policy $\hat{\pi}$ using the Regularized Actor Critic framework with entropy as regularization ($\mathcal{R}^1$) and the reward derived from GAN-GCL. To avoid undue influence from the entropy term in the regularized objective function used for policy optimization, we remove it from the retrieved GAN-GCL reward function (Eq. 6):

$$\hat{r}_\psi(s, a) = \log(\frac{D_\psi(s, a)}{1 - D_\psi(s, a)}) + \log \pi(a|s). \tag{20}$$

4.4. Model-Based Regularized Actor Critic with GAN-GCL as Guidance

The learning process of our algorithm is visualized in Fig 1 and summarized in Algorithm 1. We maintain two RL models in the training process. The first is the guidance model and it is trained using GAN-GCL reward with $\mathcal{R}^1$ regularization. The second is the base model that produces the final policy and it is trained using heuristic reward with $\mathcal{R}^2$ regularization calculated using a guidance policy. In each training iteration, the algorithm starts with generating simulated treatment trajectories. One major difference here with GAN-GCL is we use the base model to interact with the environment instead of the guidance policy learned using the GAN-GCL reward. The reason is that GAN-GCL seeks to imitate expert demonstration and the trajectory will appear to be similar to expert demonstration if we use the guidance policy to interact. This prevents the base policy from exploring the environment. After generating simulated trajectories, we sample a batch of simulated trajectories and a batch of trajectories from historical data. They are labeled correspondingly and used to update the discriminator model in GAN-GCL. Then, we combine these two batches and label each time step of using GAN-GCL reward and heuristic reward respectively. The combined batch is used to update the base model and the guidance model. Mixing real data and simulation data can help reduce the potential incorrect dynamic in the BNN model.

5. Experiments

Here we present the experiment setting and results of different treatment modeling methods.

5.1. Data

The data source for our experiments is the MIMIC-III database (Johnson et al. (2016)). It is a freely-accessible database containing de-identified ICU patient data for 53,423 distinct hospital admissions from 2001 to 2012. Following established procedures (Raghu et al. (2017)), we extracted 19443 trajectories that satisfy the sepsis-3 criteria (Singer et al. (2016)). Each trajectory contains sequential data for a patient's treatment from 24 hours before diagnosis to 48 hours after at 4-hour intervals. Patient state is represented by a vector with 48 normalized features that describe the patient's demographic and physiologic status. Treatment actions are the dose of medical intervention on the scale of $[0, 1]$, specifically intravenous (IV) fluid and vasopressor. Trajectory terminals are labeled success or failure depending on whether the patient survived 90 days after admission.

5.2. Algorithm Comparison

We compare several treatment policies to examine their quantitative performance and difference in dose distribution under different sepsis severity levels. Each parameterized policy is trained, validated, and tested using a $70 : 10 : 20$ data split. Unless otherwise specified, the reward function is $+100$ for survival, -100 for deceased and 0 for other states. In addition, the default training data is the historical data extracted from MIMIC-III. We first include the following baseline policies:

- **supervised**: It is learned by directly minimizing the negative log likelihood of the expert treatment dose.
- **DDQN**(Raghu et al. (2017)): DDQN models discretized treatment doses and is trained using the same setting as in the original paper.
- **DDPG**(Lillicrap et al. (2015)): It is a deterministic policy framework. The probability density function is computed with the output action as the mean value and $5e^{-2}$ as variance.

Algorithm 1: Model Based Regularized Actor Critic with GAN-GCL as guidance

Data: Environment models *Env*, Historical Data D_{demo}, Heuristic designed reward r
Initialize $\mathcal{R}^1$ regularized $\hat{\pi}^1_{\hat{\phi}}$ and $\hat{Q}^1_{\hat{\theta}}$, $\mathcal{R}^2$ regularized π^2_ϕ and Q^2_θ, discriminator D_ψ
$D_{sim} \leftarrow \emptyset$ `// Initialize empty pool for simulation`
for $i \leftarrow 1$ *to* N **do**
  for $j \leftarrow 1$ *to* M **do** `// generate simulated trajectories`
    $t \leftarrow 0$
    $(s'_t, a'_t) \leftarrow (s^i_0, a^i_0) \sim D_{demo}$ `// Sample initial state`
    $\tau_j \leftarrow \{(s'_t, a'_t)\}$
    repeat
      $\Delta s'_t, \varrho_t \leftarrow Env(s'_t, a'_t)$
      $s'_{t+1} \leftarrow s'_t + \Delta s'_t$
      $a'_{t+1} \sim \pi^2_\phi(\cdot|s'_{t+1})$
      $\tau_j \leftarrow \tau_j \cup \{(s'_{t+1}, a'_{t+1})\}$
      $t \leftarrow t + 1$
    until $\varrho_t \vee t > t_{\max}$
    $D_{sim} \leftarrow D_{sim} \cup \{\tau_j\}$
  end
  $\hat{D}_{sim} \sim D_{sim}, \hat{D}_{demo} \sim D_{demo}$ `// Sample training data`
  for $j \leftarrow 1$ *to* K **do** `// update discriminator model`
    $\psi \leftarrow \psi - \nabla_\psi J_D(\psi; \hat{D}_{demo}, \hat{D}_{sim})$
  end
  $\hat{D}_{train} \leftarrow \hat{D}_{sim} \cup \hat{D}_{demo}$
  label $\hat{D}_{train}$ with $\hat{r}$ using equation 20
  label $\hat{D}_{train}$ with r using heuristic reward function
  for $j \leftarrow 1$ *to* P **do** `// update base RL model and guidance RL model`
    $\hat{\phi} \leftarrow \hat{\phi} - \nabla_{\hat{\phi}} J_{\hat{\pi}^1}(\hat{\phi}; \hat{D}_{train}, \hat{r})$ `// $R^1$ regularized`
    $\hat{\theta} \leftarrow \hat{\theta} - \nabla_{\hat{\theta}} J_{\hat{Q}^1}(\hat{\theta}; \hat{D}_{train}, \hat{r})$ `// $R^1$ regularized`
    $\phi \leftarrow \phi - \nabla_\phi J_{\pi^2}(\phi; \hat{D}_{train}, r)$ `// $R^2$ regularized with $\hat{\pi}^1_{\hat{\phi}}$ as guidance`
    $\theta \leftarrow \theta - \nabla_\theta J_{Q^2}(\theta; \hat{D}_{train}, r)$ `// $R^2$ regularized with $\hat{\pi}^1_{\hat{\phi}}$ as guidance`
  end
end

- **SAC**(Haarnoja et al. (2018)): It is equivalent to the regularized Actor-Critic described in Section 4.3 with R^1 regularization.
- **GAN-GCL**(Fu et al. (2017)): It is learned using GAN-GCL with BNN as simulator.

We also conduct an ablation study on our approach to examine each component's individual effect:

- **RAC-GCL**: Policy learned using regularized Actor-Critic described in Algorithm 1 with GAN-GCL as guidance $\hat{\pi}$.
- **RAC-sup**: It is learned using regularized Actor-Critic described in Section 4.3 and R^2 regularization with a supervised policy $\hat{\pi}$ as guidance. It shows the effect of simple guidance in training using historical data.
- **SAC-model**: It is trained using the same setting as SAC but with mixed historical and simulated data from BNN. It demonstrates the effect of training on simulation data.
- **GAN-GCL+TR**: It is learned using GAN-GCL with the addition of a terminal reward $+100$ for success and -100 for failure.
- **RAC-GCL+Det**: It is the same as RAC-GCL except with a deterministic environment model.

5.3. Continuous Modeling of Treatment Doses

We first compare discretized and continuous treatment dose modeling. Figure 2 illustrates an example output of treatment dose likelihood in the discrete setting for one single treatment interval from a supervised policy and DQN policy. Each cell is associated with a likelihood value and the policy will select the highest one. We use the predicted Q-value as the likelihood for DQN. As shown in Figure 2, both the supervised policy and DQN policy have high variance in likelihood that violates the ordinal relationship of modeled doses. It is hard for a clinician to interpret why a treatment dose distant from the maximum one has a high likelihood value.

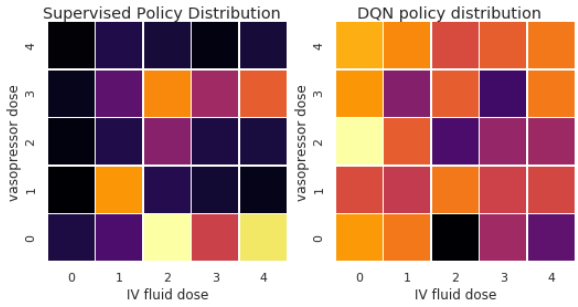

Figure 2: Likelihood distribution of discretized treatment dose modeling for a single state.

Continuous dose modeling with a multivariate truncated normal distribution produces a more explainable result. The output is the mean and standard deviation of a predicted dose distribution. Figure 3 contains a visualization of an example of a predicted dose distribution for the same treatment interval through kernel density estimation. It is more stable compared to the discrete setting. Clinicians can use mean value as suggested dose and standard deviation as the confidence of the model's recommendation.

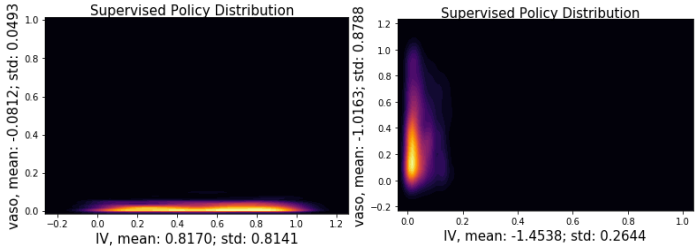

Figure 3: Likelihood distribution of continuous treatment dose modeling for a single state.

5.4. Treatment Dose Distribution Analysis

We plot the kernel density estimation for different continuous policy distributions for all treatment intervals in the test data to analyze the suggested dose pattern under different sepsis severity levels. We divide cases into Low/Medium (LM) and High (H) groups (Singer et al. (2016)) based on SOFA score. The x-axis and y-axis correspond to the advised dose of IV fluid and vasopressor respectively.

Figure 4 shows the supervised policy distribution as a generalization of expert demonstrations. In LM cases (left), the policy would recommend almost no vasopressor. Meanwhile, the dose of vasopressor increases as symptoms deteriorate (H cases, right) but the variance also increases.

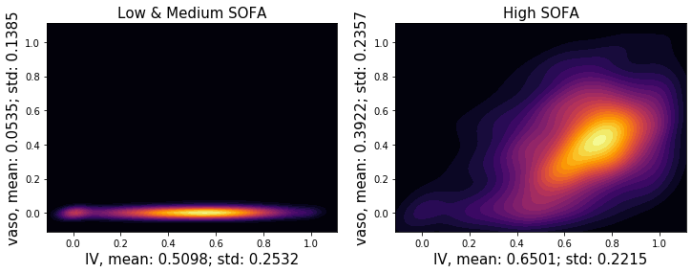

Figure 4: Supervised policy distribution across all states.

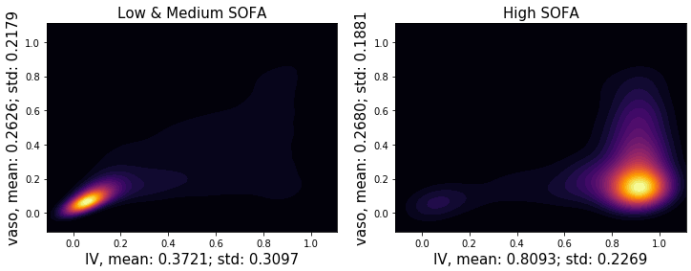

Figure 5: SAC policy distribution across all states.

Figure 5 shows plots for the SAC policy. Comparing to the supervised policy, while the majority of suggested vasopressor is still close to 0 in LM cases, the mean and standard deviation are dangerously higher. In addition, the amount of IV fluid suggested for LM patients is greatly reduced. In H cases, the SAC policy tends to suggest a high volume of IV fluid but lower volume of vasopressor.

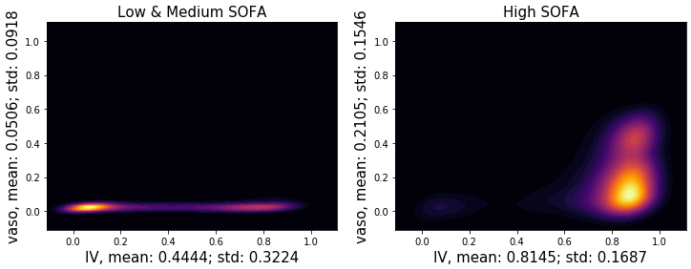

Figure 6: RAC-sup policy distribution across all states.

In the RAC-sup policy (Figure 6), usage of the supervised policy as the guidance policy reduces the amount of vasopressor suggested in LM cases and lowers the variance as well. But the policy still fails to consistently suggest high vasopressor volume needed in H cases.

Combining historical and simulation data for training (Figure 7), SAC is able to suggest a high volume of vasopressor in H cases. However, the dangerous treatment pattern in LM cases remains.

GAN-GCL (Figure 8) illustrates a similar policy distribution compared to the supervised model but with lower variance. This confirms GAN-GCL's ability to imitate expert demonstrations.

Figure 9 shows the RAC-GCL policy. Similar to RAC-sup, using a guidance policy reduces the mean and variance of the suggested vasopressor dose in LM cases. However, RAC-GCL produces

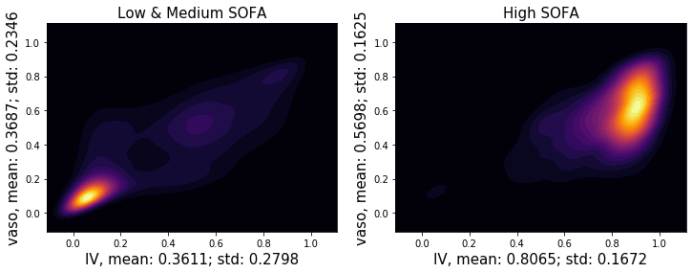

Figure 7: SAC-model policy distribution across all states.

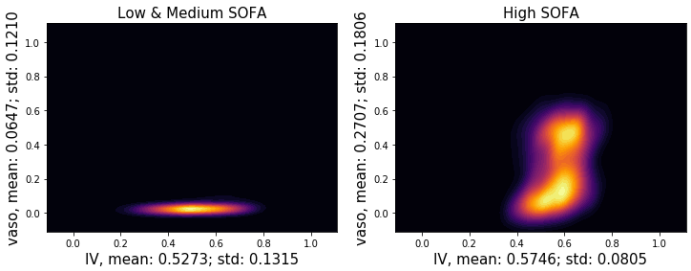

Figure 8: GAN-GCL policy distribution across all states.

a spread distribution of IV fluid in LM cases, instead of concentrating in the low IV fluid region. Furthermore, the distribution of vasopressor in H cases shifts toward the higher volume region. RAC-GCL+Det has a intermediate policy distribution between RAC-sup and RAC-GCL.

We omit the plot for DDPG and GAN-GCL+TR. DDPG is only capable of recommending extreme doses, either 0 or 1. GAN-GCL+TR shows significant similarity with GAN-GCL, suggesting a simple addition to combine two reward functions may not be sufficient.

5.5. Quantitative Analysis

To quantify performance, we compute an evaluative Q-value $Q_E(s, a)$ and V-value $V_E(s)$ for each policy $\gamma = 1$, terminal reward $r'_T = 1$ if the patient survived and $r'_t = 0$ for all other time steps. To avoid error from the BNN influencing the results, we only use historical data for evaluation. We include an importance sampling ratio $\rho_t = \frac{\pi(a_t|s_t)}{\pi_b(a_t|s_t)}$ (Sutton and Barto (2018)) to learn accurate policy values for a reward sparser than the one in the policy training.

$$Q_E(s_t, a_t) = r'_t + \gamma V_E(s_{t+1}), \tag{21}$$

$$V_E(s_t) = r'_t + \rho_t * V_E(s_{t+1}). \tag{22}$$

π_b is the behavior policy that generated treatment trajectories in the historical data. We use a supervised model to estimate π_b. Q_E and V_E allows us to compute performance measures excluding the entropy or cross entropy component in the regularized Q-value.

We design such a setting for Q_E and V_E computation to account for patients' survival only and satisfy the condition of applying Off-Policy Classification (OPC) and Soft OPC (SOPC) scores (Irpan et al. (2019)). Both of them measure the difference of a decision function between success

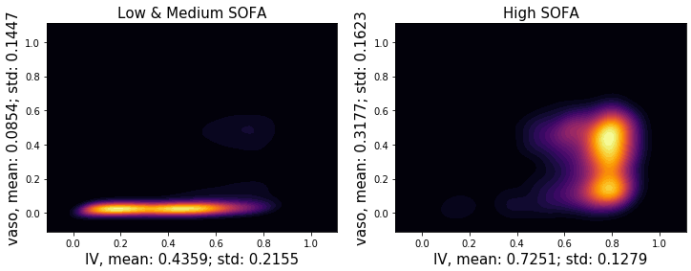

Figure 9: RAC-GCL policy distribution across all states.

trajectories (survival) and overall trajectories. We label all time steps of successful trajectories as positive $y = 1$ and compute these scores using the $Q_E(s, a)$ of each policy.

$$OPC(Q_E) = p(y = 1)\mathbb{E}_{y=1}[\mathbb{1}_{Q_E(s,a)>b}] - \mathbb{E}[\mathbb{1}_{Q_E(s,a)>b}], \tag{23}$$

$$SOPC(Q_E) = p(y = 1)\mathbb{E}_{y=1}[Q_E(s, a)] - \mathbb{E}[Q_E(s, a)]. \tag{24}$$

$\mathbb{1}_{Q(s,a)>b}$ is the indicator function. Found by line search, b is the threshold that maximizes OPC.

We also compute the Weighted Doubly Robust (WDR) estimator (Thomas and Brunskill (2016)) as the main performance measurement

$$WDR(D) = \sum_i \sum_t \gamma^t w_t^i r_t'^i - \sum_i \sum_t \gamma^t (w_t^i Q_E(s_t^i, a_t^i) - w_{t-1}^i V_E(s_t^i)), \tag{25}$$

where i is the index for the trajectory and t is the index of the timestep in each trajectory. $w_t^i = \rho_t^i / \sum_j \rho_t^j$ is the weighted importance sampling ratio.

Finally, we calculate the RMSE to characterize similarity with expert demonstrations and show the effect of using a guidance policy. We also report the Pearson correlation coefficient of each treatment dose with the Sequential Organ Failure Assessment (SOFA) score (Singer et al. (2016)) to show the statistical correlation between treatment and the severity of the sepsis symptoms. We report the means of selected metrics over 20 trials.

Table 1: Quantitative results of different models.

	WDR	OPC×10	SOPC×10	SOFA-IV	SOFA-Vaso	RMSE
supervised	0.6778	1.2459	0.5064	0.0610	0.4361	**0.2337**
DDQN	0.6424	1.0770	0.4557	0.0227	0.2187	0.3778
DDPG	0.6727	1.2394	0.4921	0.0129	0.1023	0.4395
SAC	0.7207	1.1894	0.4895	0.3568	0.1536	0.3873
GAN-GCL	0.8491	1.1730	0.4796	0.0604	0.4359	0.2383
SAC-model	0.8286	1.1816	0.5043	0.3718	0.2771	0.3738
RAC-sup	0.7911	**1.2758**	0.5134	0.3092	0.4162	0.2710
GAN-GCL+TR	0.7366	1.1869	0.4924	0.1349	0.4458	0.2495
RAC-GCL+Det	0.8674	1.2452	0.5233	0.2258	0.4406	0.2570
RAC-GCL	**0.8912**	1.2746	**0.5352**	0.3202	0.4529	0.2652

Table 1 shows selected metrics for different policies. An ideal treatment policy should have higher WDR, OPC and SOPC while maintaining a low RMSE. We observe that the difference in

terms of OPC and SOPC are not significant across different policies. Because OPC and SOPC are studied for deterministic environments, a stochastic and partially-observable environment may limit their capability in performance measure.

Comparing continuous probablistic modeling to other baselines, SAC achieves improved performance but notably higher RMSE than supervised policy. GAN-GCL results in a further improved WDR and reduced RMSE. This could be the combined effect of GAN-GCL reward and training on simulation data since SAC-model also improves performance, though its RMSE remains high. However, simply adding GAN-GCL reward to designed terminal reward reduces the performance. Considering R^2 regularization, both RAC-sup and RAC-GCL increases performance measures while moderate the high RMSE of SAC. This suggests the effectiveness of guidance regularization. GAN-GCL can be more desirable as guidance because of RAC-GCL's better performance. Finally, higher scores of RAC-GCL than RAC-GCL+Det suggests that a stochastic simulation can be more realistic to the environment and reduce the effect of model overfitting.

An distinguishing property of learnt policies is the correlation of doses with SOFA score. For supervised and GAN-GCL policy, SOFA score has a negligible correlation with IV fluid doses but a significant one with vasopressor. This is medically explainable since IV fluid can be applied under diverse situations that may not relate to SOFA condition. On the other hand, vasopressor is usually only prescribed to patients with severely low blood pressure that is common in high SOFA situation. However, SAC, RAC-sup and RAC-GCL all produce a considerable correlation between suggested IV fluid and SOFA score. Whether such phenomenon is desirable requires further investigation.

Overall, RAC with R^2 regularization using a guidance policy demonstrates improved performance with reduced RMSE. This suggests that including the guidance policy as a constraint can regularize the policy to pick treatments closer to expert demonstrations, and dynamically adjusting the temperature parameter controls the extent of the regularization to ensure the policy is optimized towards the maximum entropy objective. RAC-GCL produces a effective combination of terminal reward and GAN-GCL reward, leading to a stronger guidance effect. It is likely the result of implicit balance between these two in RAC-GCL. Updated by regularized Actor-Critic, the guidance policy contains the retrieved GAN-GCL reward in its probability density. A dynamic temperature parameter that automatically balances the influence of guidance can consequently produce an indirect balance between terminal reward and IRL reward.

6. Conclusion

In this paper, we utilize a regularized Actor-Critic framework to recommend continuous treatment doses. By modeling doses with a truncated, multivariate normal distribution, our approach can perform continuous treatment dose recommendation with a confidence level. It is more suitable for interpretation and thus facilitates clinicians' decision making. We explore different techniques to alleviate the challenge of off-line learning for RL to derive an effective treatment strategy. Combining these methods, we propose a model-based regularized actor critic framework with GAN-GCL as a guidance policy. Our framework utilizes a simulation model to assist in exploring the environment while using KL-divergence to incorporate the influence of a guidance policy. Experimental results suggest that such a combination can significantly improve model performance while retaining a safe similarity to clinician treatments. In addition, a RL policy learned from an IRL derived reward (*e.g.* GAN-GCL) is shown to be more effective as guidance. Future work includes a more detailed analysis of learned policy distributions for medical explanation and exploration of other guidance.

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 182–189, 2011.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *arXiv preprint arXiv:1710.11248*, 2017.

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

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

Alex Irpan, Kanishka Rao, Konstantinos Bousmalis, Chris Harris, Julian Ibarz, and Sergey Levine. Off-policy evaluation via off-policy classification. *arXiv preprint arXiv:1906.01624*, 2019.

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.

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019.

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, 2018.

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 11761–11771, 2019.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

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.

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.

Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49 (2-3):267–290, 2002.

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.

Niranjani Prasad, Li-Fang Cheng, Corey Chivers, Michael Draugelis, and Barbara E Engelhardt. A reinforcement learning approach to weaning of mechanical ventilation in intensive care units. *arXiv preprint arXiv:1704.06300*, 2017.

Aniruddh Raghu, Matthieu Komorowski, Imran Ahmed, Leo Celi, Peter Szolovits, and Marzyeh Ghassemi. Deep reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1711.09602*, 2017.

Aniruddh Raghu, Matthieu Komorowski, and Sumeetpal Singh. Model-based reinforcement learning for sepsis treatment. *arXiv preprint arXiv:1811.09602*, 2018.

Mervyn Singer, Clifford S Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R Bernard, Jean-Daniel Chiche, Craig M Coopersmith, et al. The third international consensus definitions for sepsis and septic shock (sepsis-3). *Jama*, 315(8):801–810, 2016.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Philip Thomas and Emma Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International Conference on Machine Learning*, pages 2139–2148, 2016.

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Thirtieth AAAI conference on artificial intelligence*, 2016.

Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2447–2456. ACM, 2018.

Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. *arXiv preprint arXiv:1803.04386*, 2018.

Chao Yu, Jiming Liu, and Hongyi Zhao. Inverse reinforcement learning for intelligent mechanical ventilation and sedative dosing in intensive care units. *BMC medical informatics and decision making*, 19(2):57, 2019.

Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. PhD thesis, figshare, 2010.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. 2008.