Adversarial Cooperative Imitation Learning for Dynamic Treatment Regimes[*]

Lu Wang
East China Normal University
luwang@stu.ecnu.edu.cn

Wenchao Yu[*]
NEC Laboratories America
wyu@nec-labs.com

Wei Cheng
NEC Laboratories America
weicheng@nec-labs.com

Martin Renqiang Min
NEC Laboratories America
renqiang@nec-labs.com

Bo Zong
NEC Laboratories America
bzong@nec-labs.com

Xiaofeng He[*]
East China Normal University
xfhe@stu.ecnu.edu.cn

Hongyuan Zha
Georgia Tech
zha@cc.gatech.edu

Wei Wang
University of California Los Angeles
weiwang@cs.ucla.edu

Haifeng Chen
NEC Laboratories America
haifeng@nec-labs.com

ABSTRACT

Recent developments in discovering dynamic treatment regimes (DTRs) have heightened the importance of deep reinforcement learning (DRL) which are used to recover the doctor's treatment policies. However, existing DRL-based methods expose the following limitations: 1) supervised methods based on behavior cloning suffer from compounding errors; 2) the self-defined reward signals in reinforcement learning models are either too sparse or need clinical guidance; 3) only *positive trajectories* (e.g. survived patients) are considered in current imitation learning models, with *negative trajectories* (e.g. deceased patients) been largely ignored, which are examples of what not to do and could help the learned policy avoid repeating mistakes. To address these limitations, in this paper, we propose the adversarial cooperative imitation learning model, ACIL, to deduce the optimal dynamic treatment regimes that mimics the positive trajectories while differs from the negative trajectories. Specifically, two discriminators are used to help achieve this goal: an *adversarial discriminator* is designed to minimize the discrepancies between the trajectories generated from the policy and the positive trajectories, and a *cooperative discriminator* is used to distinguish the negative trajectories from the positive and generated trajectories. The reward signals from the discriminators are utilized to refine the policy for dynamic treatment regimes. Experiments on the publicly real-world medical data demonstrate that ACIL improves the likelihood of patient survival and provides better dynamic treatment regimes with the exploitation of information from both positive and negative trajectories.

KEYWORDS

imitation learning, dynamic treatment regimes, generative adversarial networks, reinforcement learning

ACM Reference Format:
Lu Wang, Wenchao Yu*, Wei Cheng, Martin Renqiang Min, Bo Zong, Xiaofeng He*, Hongyuan Zha, Wei Wang, and Haifeng Chen. 2020. Adversarial Cooperative Imitation Learning for Dynamic Treatment Regimes. In *Proceedings of The Web Conference 2020 (WWW '20), April 20–24, 2020, Taipei, Taiwan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3366423.3380248

1 INTRODUCTION

A dynamic treatment regime (DTR) is a sequence of tailored treatment decision rules that specify how the treatments should be adjusted through time according to the dynamic states of patients [5, 24]. Each rule takes input information, e.g. medical history, laboratory measurements, demographics, etc., of the patients, and recommends treatment options which aim to optimize the effectiveness of the treatment program.

Motivated by the remarkable success of deep reinforcement learning (DRL) in finding effective dynamic policies that can be applied on areas like economics [18, 38], transportation [34] and robotics [14], a set of studies have focused on using DRL to learn the optimal dynamic treatment regimes from electronic health records (EHRs) [3, 21, 28, 31, 35, 37]. Given these observational medical data, the optimal DTR estimation is to learn a policy guided by rewards, e.g., a negative reward is given to a patient who died in-hospital and a positive reward to someone who is discharged. Existing methods can be generally divided into two categories: behavior cloning (BC) and reinforcement learning with self-defined reward functions. Supervised learning methods based on BC can effectively recover the doctor's treatment policies with abundant health data [2, 7, 17, 37]. However, BC suffers from compounding errors [30], because the agent greedily mimics the demonstrated actions, and error accumulates as the policy unrolled [9, 29]. In the EHRs, some patients are cured but there are still a certain amount of unhealed patients. Several studies consider these EHRs as sub-optimal trajectories and use reinforcement learning to infer optimal decisions from sub-optimal training examples with the manually designed reward functions [3, 21, 28, 35]. However, the reward signals are extremely sparse under this setting, which introduces credit assignment problem, where immediate rewards are almost zero, and it's hard to identify which actions are useful in obtaining the final feedback. Though the clinically guided reward functions can help provide

[*] Corresponding authors

WWW '20, April 20–24, 2020, Taipei, Taiwan

ACM ISBN 978-1-4503-7023-3/20/04.
https://doi.org/10.1145/3366423.3380248

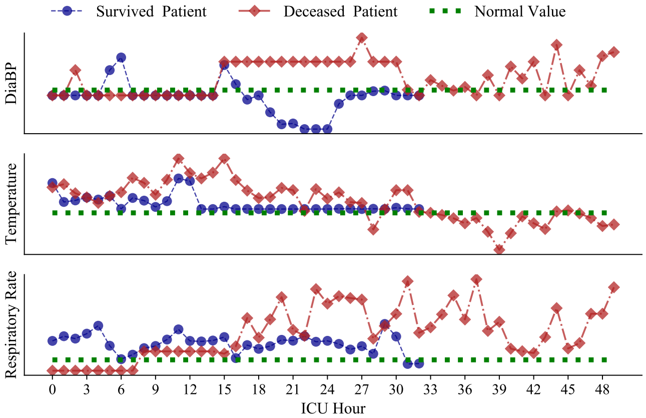


Figure 1: Two Sepsis patients extracted from MIMIC-III with similar initial lab test results (i.e., DiaBP, temperature, respiratory rate) and demographics but different treatments. The patient with blue curve was survived while the patient with red curve deceased.

dense reward signals [27], it requires expert knowledge and cannot be easily transferred to different domains.

Recent developments in imitation learning, such as inverse reinforcement learning (IRL) [1, 40] and adversarial imitation learning (AIL) [9, 15, 22] avoid the compounding errors by training an agent to match the demonstrations over a long horizon with an explicit or implicit defined reward function. Meanwhile, the learned reward function also alleviates the sparse reward problem. Thus IRL and AIL are potentially suitable for discovering the optimal dynamic treatment regimes. We define *positive trajectories* as therapeutic process of patients with positive outcomes (i.e., cured or survived patients) and *negative trajectories* as patients with negative outcomes (i.e., deceased patients). It can be seen that these imitation learning methods only consider the positive trajectories are optimal and learn a policy to recover these trajectories. The information in the negative trajectories has been largely ignored, which could potentially help the learned policy to avoid repeating mistakes. As illustrated in Fig. 1, two Sepsis patients with similar initial states result in different outcomes when taking different treatment plans. Thus it's essential to learn the optimal dynamic treatment regime that matches the positive demonstrations while differs from the negative trajectories. In other words, the negative trajectories which results in death can also guide the policy learn what not to do.

To address the aforementioned limitations, in this paper, we propose the adversarial cooperative imitation learning model (short for ACIL) to learn the optimal dynamic treatment regimes. As shown in Fig. 2, ACIL learns the optimal DTR policy by taking positive trajectories and negative trajectories as inputs. The input trajectories pass through two discriminators to help the policy mimics the positive trajectories while stays far away from the negative trajectories. To achieve this goal, ACIL consists of an adversarial discriminator and a cooperative discriminator to learn the policy. The adversarial discriminator is trained to minimize the divergence between the learned policy and the positive trajectories, while the cooperative discriminator is trained to distinguish the positive trajectories (including trajectories generated from the learned policy) and negative trajectories. ACIL utilizes the reward signals from the discriminators to help refine the policy for dynamic treatment regimes and the patient model (act as *environment*) built with variational autoencoders. We quantitatively validate the effectiveness of ACIL on real-world medical data, which demonstrates the effectiveness of the proposed model. To summarize, the main contributions are as follows:

- We propose a novel adversarial cooperative imitation learning model, ACIL, to learn optimal dynamic treatment regime policies, which includes an adversarial discriminator and a cooperative discriminator to better exploit the information from the positive and negative trajectories.
- The environment is simulated with the patient model, which leverages the variational autoencoder architecture to take the current *state* and *action* values as inputs and output the successor *state*.
- Quantitative experiments and qualitative case studies on MIMIC-III demonstrate that ACIL reduces the estimated mortality and provides better dynamic treatment regimes with the usage of all treatment demonstrations.

The rest of this paper is organized as follows. Section 2 introduces the preliminaries of IRL and AIL. Section 3 describes the ACIL model and provides theoretical analysis of the proposed model. Section 4 empirically evaluates ACIL on MIMIC-III dataset. We summarize the related work in Section 5, followed by the conclusions in Section 6.

2 PRELIMINARIES

Generally, given a set of experts' demonstration trajectories τ, which consists of sequences of states and actions $(s_0, a_0, s_1, a_1, \dots)$ drawn from the expert policy π, the goal of imitation learning is to learn a policy $\pi_\theta(a|s)$ which can replicate experts' behaviors. The imitation learning methods can be generally grouped into three categories: behavior cloning (BC), inverse reinforcement learning (IRL) and adversarial imitation learning (AIL).

2.1 Behavior Cloning

BC aims to learn the policy $\pi_\theta(a|s)$ via supervised learning. Given the fixed action space or classes, BC learns a policy mapping from states to experts' actions with the tuple datasets $\{(s_0, a_0), (s_1, a_1), \dots\}$,

$$\arg\min_\theta \mathbb{E}_{(s,a)\sim P^*} L(a, \pi_\theta(s)), \tag{1}$$

where $P^* = P(s|\pi^*)$ is the distribution of states visited by expert. Due to the standard i.i.d. assumption in the supervised learning, the errors induced by BC are compounding over the length of the trajectories.

2.2 Inverse Reinforcement Learning

In inverse reinforcement learning, we aim to learn the reward function based on the expert demonstrations. The reward function can be considered as a linear combination of features,

$$R(s) = \omega^\intercal f(s), \tag{2}$$

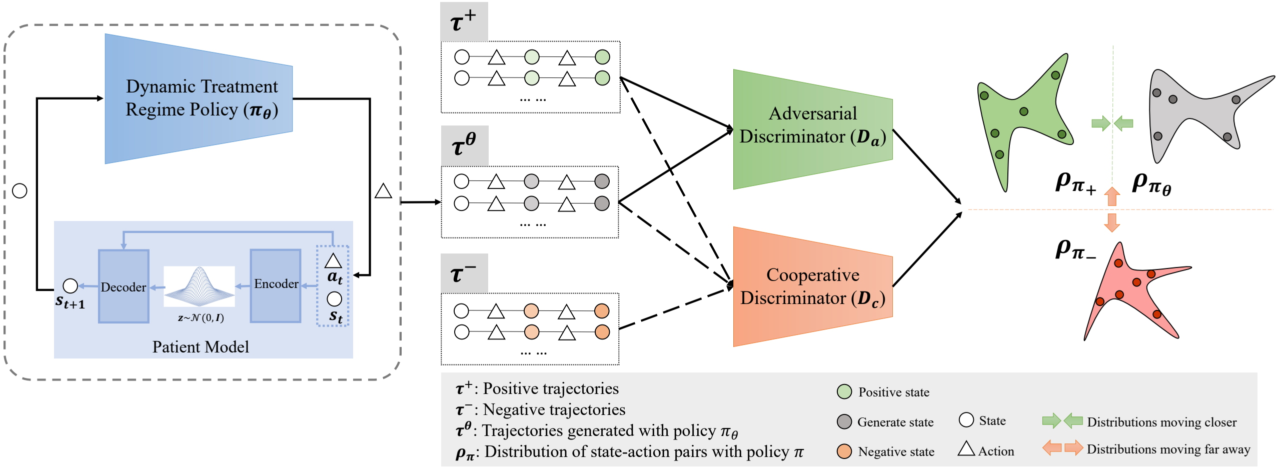

Dynamic Treatment Regime Policy (π_θ)
Decoder
Encoder
$z \sim \mathcal{N}(0, I)$
a_t
s_t
s_{t+1}
Patient Model
τ^+
τ^θ
τ^-
… …
Adversarial Discriminator (D_a)
Cooperative Discriminator (D_c)
ρ_{π_+}
ρ_{π_θ}
ρ_{π_-}
τ^+: Positive trajectories
τ^-: Negative trajectories
τ^θ: Trajectories generated with policy π_θ
ρ_π: Distribution of state-action pairs with policy π
Positive state
Generate state
Negative state
State
Action
Distributions moving closer
Distributions moving far away

Figure 2: Illustration of the ACIL model for learning optimal dynamic treatment regimes

Table 1: Notation descriptions

Notation	Description
$s_t \in \mathcal{S}$	The state which consists of the patients' demographics and clinical variables.
$a_t \in \mathcal{A}$	The action, $\mathcal{A} \in \{0,1\}^K$, the k^{th} dimension of a_t in $\{0,1\}$ indicates whether the k^{th} medication or medication dosage is chosen for a patient.
$\pi_E = \{\pi_+, \pi_-\}$	The behavior policy which consists of a positive policy π_+ and a negative policy π_-.
$\rho_\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$	The distribution of state-action pairs that the policy π interacts with the environment, $\rho_\pi(s,a) = \pi(a\|s) \sum_{t=0}^{T} \gamma P(s_t = s\|\pi)$, where γ is the discounting factor, and the successor states are drawn from $P(s\|\pi)$.
$\tau^+ = (s_1^+, a_1^+, \ldots)$	Positive trajectories generated by π_+ which consists of sequences of states and actions.
$\tau^- = (s_1^-, a_1^-, \ldots)$	Negative trajectories generated by π_-.
$\pi_\theta(a\|s)$	The learned policy with parameter vector θ.
$\tau^\theta = (s_1, a_1, \ldots)$	The trajectories generated by π_θ.
D_a	The *adversarial* discriminator.
D_c	The *cooperative* discriminator.
E_{w_1}	Encoder with parameter w_1 in patient model.
D_{w_2}	Decoder with parameter w_2 in patient model.
G_w	$G_w = \{E_{w_1}, D_{w_2}\}$. Patient model with an encoder and a decoder.

where $\omega \in \mathbb{R}^n$ is the weight vector required to be learned, $f : \mathcal{S} \to \mathbb{R}^n$. The value function of policy π_θ can be expressed as,

$$\begin{aligned} V_{\pi_\theta}(s) &= \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t)|s_0 = s\right] \\ &= \omega^\top \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t f(s_t)|s_0 = s\right] = \omega^\top \mu(\pi_\theta), \end{aligned} \quad (3)$$

where $\mu(\pi_\theta) \in \mathbb{R}^n$ is the discounted weighted frequency of state features $f(s)$ under policy π_θ. With the assumptions on the optimality of the demonstrations, IRL learns ω under the constrains that:

$$\mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|s_0 = s\right] \geq \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R^*(s_t)|s_0 = s\right], \ \forall \pi_\theta, \quad (4)$$

where R^* indicates the optimal reward function, and π^* is the expert policy. With Eq. (4), the value of a policy expressed as,

$$\omega^\top \mu(\pi^*|s_0 = s) \geq \omega^\top \mu(\pi_\theta|s_0 = s), \quad (5)$$

which indicates IRL tries to find the reward function such that the expert policy outperforms other policies. Thus we need to find the π_θ, such that $|\mu(\pi_\theta|s_0 = s) - \mu(\pi^*|s_0 = s)| \leq \epsilon$. This is equivalent to match the discounted state visitation features to the expert [1],

$$|\omega^\mathrm{T} \mu(\pi_\theta|s_0 = s) - \omega^\mathrm{T} \mu(\pi^*|s_0 = s)| \leq \epsilon, \ \forall \|\omega\|_\infty \leq 1. \quad (6)$$

This observation leads to learn a policy π_θ that is as good as the expert policy while $\gamma \leq \epsilon/2$.

2.3 Adversarial Imitation Learning

Instead of indirectly learning the policy π_θ as IRL, adversarial imitation learning directly learns π_θ by minimizing the Jensen-Shannon divergence between expert' policy π_E and the learned policy π_θ

$$D_{JS}(\rho_{\pi_\theta}, \rho_{\pi_E}) = D_{KL}(\rho_{\pi_\theta}|\frac{\rho_{\pi_\theta} + \rho_{\pi_E}}{2}) + D_{KL}(\rho_{\pi_E}\|\frac{\rho_{\pi_\theta} + \rho_{\pi_E}}{2}),$$

where the occupancy measure ρ_π is the distribution of state-action pairs that the policy π interacts with the environment (see Table 1 for details). AIL utilizes a generative adversarial network to minimize the Jensen-Shannon divergence via a generator π_θ and a discriminator $D(\cdot)$ with the following objective function:

$$\max_{D \in (0,1)^{\mathcal{S} \times \mathcal{A}}} \mathbb{E}_{\rho_{\pi_\theta}}[\log(D(s,a))] + \mathbb{E}_{\rho_{\pi_E}}[\log(1 - D(s,a))], \quad (7)$$

where $\mathcal{S}$ is the state set, and $\mathcal{A}$ is the action set, as defined in Table 1.

3 METHOD

Problem Definition: Given two set of demonstration trajectories, namely, the positive trajectories $\tau^+ = \{\tau_1^+, \tau_2^+, ..., \tau_m^+\}$ and the negative trajectories $\tau^- = \{\tau_1^-, \tau_2^-, ..., \tau_n^-\}$, which are generated from behavior policies $\pi_E = \{\pi_+, \pi_-\}$. Positive trajectories are the demonstrations that can achieve the goal of the task (e.g. doctors' treatment policies for cured patients, experts' driving trajectories, etc.), while negative trajectories result in a failure outcome. Each trajectory τ consists of a sequence of state-action pairs (s_t, a_t), i.e., $\tau^+ = (s_0^+, a_0^+, s_1^+, a_1^+, ...)$. Our goal is to learn a policy π_θ which can recover positive trajectories while differs from the negative trajectories.

3.1 The ACIL Model

The goal of ACIL is to learn a policy π_θ from doctors' prescriptions, including positive trajectories as therapeutic process of patients with positive outcomes, and negative trajectories with bad outcomes (e.g. deceased patients), without interacting with the experts and reward signals. The learned policy are enforced to mimic the positive trajectories while staying far away from the negative trajectories. To achieve this objective, ACIL consists of an adversarial discriminator D_a, a cooperative discriminator D_c and a patient model G_w to learn the policy π_θ, as illustrated in Fig. 2. The adversarial discriminator D_a is trained to minimize the Jensen–Shannon (JS) divergence between the distributions of state-action pairs ρ_{π_θ} and ρ_{π_+}, which are generated by interacting with the environment using policy π_θ and the expert policy π_+, respectively. Meanwhile, the cooperative discriminator D_c is trained to distinguish the positive trajectories (τ^θ, τ^+) from the negative trajectories τ^-, which is equivalent to maximize the JS divergence between $(\rho_{\pi_\theta}, \rho_{\pi_+})$ and ρ_{π_-} (as shown in Proposition 3.2). ACIL utilizes the feedback signals from the discriminators to help refine the policy π_θ for dynamic treatment regimes. The patient model G_w, acts as an environment simulator, provides the model dynamics $P(s|\pi)$ where the successor states are drawn from.

3.1.1 *Patient Model as An Environment Simulator.*

The environment can be simulated with generative models such as variational auto-encoder (VAE) [20] and GAN [12] for model-based reinforcement learning [4, 6] and trajectory embedding [8]. Here we leverage the state-action conditioned VAE architecture to build the patient model G_w which transforms the state distribution into an underlying latent space. In detail, the patient model consists of an encoder E_{w_1}, which maps the current states s_t, action a_t to the latent distribution $z \sim \mathcal{N}(\mu, \sigma)$, and a decoder D_{w_2}, which maps latent z and current state s_t and action a_t into the successor state $\hat{s}_{t+1}$. The goal to train the patient model is to minimize the error between s_{t+1} and $\hat{s}_{t+1}$ under the latent distribution z. The objective function can be expressed as,

$$\min_{w} \sum_{(s_t, a_t, s_{t+1})} \|s_{t+1} - \hat{s}_{t+1}\|_2 + \alpha D_{KL}(\mathcal{N}(\mu, \sigma) \| \mathcal{N}(0, 1)), \tag{8}$$

where $\mu, \sigma = E_{w_1}(s_t, a_t)$, and $\hat{s}_{t+1} = D_{w_2}(s_t, a_t, z)$. After we obtain the well-trained patient model, the state transitions can be predicted using $D_{w_2}(s_t, a_t, z)$, with current state s_t, action a_t and z as inputs.

3.1.2 *Adversarial Cooperative Imitation Learning.*

In practice, we compare the difference between the π_θ and π_+ via their generated trajectories. For a policy $\pi \in \Pi$, its occupancy measure $\rho_\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is defined as $\rho_\pi(s, a) = \pi(a|s) \sum_{t=0}^{T} \gamma P(s_t = s|\pi)$, which can be interpreted as the distribution of state-action pairs that the policy interacts with the environment. Multiple-layer perceptrons (MLPs) are used for π_θ, D_a and D_c. π_θ takes the state of the patients as inputs, and returns the recommended medications. $D_a(s, a)$ presents the probability that state-action pair (s, a) comes from π_+. $D_c(s, a)$ indicates the probability that (s, a) is belong to the positive demonstration (π_+ and π_θ). We train D_a and π_θ adversarially. Simultaneously, we train D_c and π_θ in a cooperative fashion. In summary, π_θ, D_a and D_c play the three-player min-max game which can be defined as follows,

$$\begin{aligned} \min_{\pi_\theta, D_c} \max_{D_a} \; & \omega_\alpha(\mathbb{E}_{\rho_{\pi_\theta}}[\log(1 - D_a(s, a))] + \mathbb{E}_{\rho_{\pi_+}}[\log(D_a(s, a))]) \\ & - \omega_\beta(\mathbb{E}_{\rho_{\pi_\theta}, \rho_{\pi_+}}[\log(D_c(s, a))] + \mathbb{E}_{\rho_{\pi_-}}[\log(1 - D_c(s, a))]) \\ & - \lambda H(\pi_\theta), \end{aligned} \tag{9}$$

where $\omega_\alpha \in [0, 1]$ and $\omega_\beta \in [0, 1]$ balance the importance between the adversarial discriminator and the cooperative discriminator. $H(\pi_\theta) \triangleq \mathbb{E}_{\pi_\theta}[-\log \pi_\theta(a|s)]$ is the causal entropy of policy π_θ which encourages the policy diversity.

Updating the Adversarial Discriminator. The adversarial discriminator $D_a : \mathcal{S} \times \mathcal{A} \rightarrow (0, 1)$ estimates the probability that a state-action pair (s, a) comes from π_+ rather than π_θ. The objective function can be described as follows,

$$\max_{D_a} \mathbb{E}_{\rho_{\pi_\theta}}[\log(1 - D_a(s, a))] + \mathbb{E}_{\rho_{\pi_+}}[\log(D_a(s, a))]). \tag{10}$$

D_a is called *adversarial* discriminator because the goals to optimizing D_a and π_θ are opposite. D_a is to minimize the probability of the state-action pair generated by π_θ, while π_θ is to maximize the probability of D_a making a mistake. This objective is equivalent to minimize the JS divergence, $D_{JS}(\rho_{\pi_\theta} \| \rho_{\pi_+})$, between ρ_{π_θ} and ρ_{π_+}.

Updating the Cooperative Discriminator. The goal of the cooperative discriminator $D_c : \mathcal{S} \times \mathcal{A} \rightarrow (0, 1)$ is to differentiate the generated samples and the positive samples from the negative samples. The objective can be expressed as,

$$\max_{D_c} \mathbb{E}_{\rho_{\pi_\theta}, \rho_{\pi_+}}[\log(D_c(s, a))] + \mathbb{E}_{\rho_{\pi_-}}[\log(1 - D_c(s, a))]. \tag{11}$$

This objective function characterizes the optimal negative log loss of classifying the positive trajectories generated from π_θ and π_+, and the negative trajectories generated from π_-. We name it *cooperative* discriminator because both the goal of D_c and π_θ are to maximize the probability of the data generated by π_θ to be positive. D_a, D_c can be considered as reward functions to help refine π_θ. When the distribution ρ_{π_θ} is different from ρ_{π_-}, it will receive a large reward form D_c. We also show that, with the optimal D_c, the loss of π_θ is $D_{JS}(\rho_{\pi_+} + \rho_{\pi_\theta} \| \rho_{\pi_-})$ (details can be found in Section 3.2).

Updating the Policy. The objective of updating π_θ is to mimic the positive trajectories, while staying "far" away from the negative samples. Under this setting, π_θ incorporates the reward signals from both D_a and D_c. The signal from D_a is used to push π_θ close to π_+, while the signal from D_c separates π_θ and π_-. The loss function is defined as,

$$\min_{\pi_\theta} \omega_\alpha(\mathbb{E}_{\rho_{\pi_\theta}}[\log(1-D_a(s,a))])-\omega_\beta(\mathbb{E}_{\rho_{\pi_\theta}}[\log(D_c(s,a))]-\lambda H(\pi_\theta), \tag{12}$$

where $H(\pi)$ is the causal entropy of the policy to encourage the diversity of the learned policy and $\lambda \geq 0$ is used to control $H(\pi_\theta)$. $\omega_\alpha, \omega_\beta \in [0, 1]$ balance these two reward signals.

When both D_a and D_c become optimal, we can show that the objective, defined in Eq. (9), is equivalent to the following optimization problem.

$$\min_{\pi_\theta} D_{JS}(\rho_{\pi_+} \| \rho_{\pi_\theta}) - D_{JS}((\rho_{\pi_+} + \rho_{\pi_\theta}) \| \rho_{\pi_-}) - \lambda H(\pi_\theta), \tag{13}$$

which finds a policy whose occupancy measure minimizes the JS divergence to π_+, and maximize the JS divergence to π_-. Section 3.2 provides the detailed proof.

The ACIL model learning is summarized in Algorithm 1. We first train patient model G_w, followed by training D_a, D_s, and π_θ iteratively.

Algorithm 1 ACIL for DTR Learning

```
Require: Positive and negative trajectories $\tau^+$ and $\tau^-$ generated
    by behavior policies $\pi_+$ and $\pi_-$, batch $\mathcal{B}$, horizon $T$, mini-batch
    size $N$, episode size $M$. Initial the parameters of $\pi_\theta$, $D_a$, $D_c$ and
    $G_w = \{E_{w_1}, D_{w_2}\}$.
for $t = 1$ to $T$ do
    Sample mini-batch of N transitions from both $\tau^+$ and $\tau^-$
    $(s_t, a_t, s_{t+1})$ from $\mathcal{B}$
    $\mu, \sigma = E_{w_1}(s_t, a_t)$, $\bar{s}_{t+1} = D_{w_2}(s_t, a_t, z)$,$z \sim \mathcal{N}(\mu, \sigma)$
    Update $w$ with Eq. (8).
end for
for episode = 0 to $M$ do
    Sample trajectories $\tau_i \sim \pi_\theta$, $\tau_i^+ \sim \pi_+$, $\tau_i^- \sim \pi_-$ ;
    Update the parameters of $D_a$ with $\tau^+$, $\tau^\theta$ and the gradient
    $\hat{\mathbb{E}}_{\tau_i^\theta}[\nabla \log(1 - D_a(s,a))] + \hat{\mathbb{E}}_{\tau_i^+}[\nabla \log(D_a(s,a))]$, where $\hat{\mathbb{E}}_\tau$
    presents the estimated expectation with $\tau$.
    Update the parameters of $D_s$ with $\tau^+, \tau^-, \tau^\theta$ and the gradient
    $\hat{\mathbb{E}}_{\tau_i^\theta, \tau_i^+}[\nabla \log(D_c(s,a))] + \hat{\mathbb{E}}_{\tau_i^-}[\nabla \log(1 - D_c(s,a))]$.
    Update the parameters of $\pi_\theta$ using TRPO with the cost func-
    tion $\omega_\alpha \log(D_a(s,a)) + \omega_\beta \log(D_s(s,a))$. Specifically, $\theta$ is up-
    dated by the gradient,
    $\hat{\mathbb{E}}_{\tau_i^\theta}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta)$,
    where $Q(s,a) = \hat{\mathbb{E}}_{\tau_i^\theta}[\omega_\alpha \log(D_a(s,a) + \omega_\beta \log(D_s(s,a))]$.
end for
```

3.2 Theoretical Analysis

We now provide formal theoretical analysis of ACIL. The policy π_θ implicitly defines a probability distribution ρ_{π_θ} of state-action pairs. We would like ACIL to coverage to the distribution of the positive trajectories ρ_{π_+}, if given enough capacity and training time. For simplicity of analysis, we set the balancing factors $\omega_\alpha, \omega_\beta$ to 1, and neglect the regularizer $H(\cdot)$. The objective function of ACIL in Eq. (9) can be rewritten as follows,

$$\begin{aligned} J_{\pi_\theta, D_a, D_c} =& \mathbb{E}_{\rho_{\pi_\theta}}[\log(1 - D_a(s,a))] + \mathbb{E}_{\rho_{\pi_+}}[\log(D_a(s,a))] \\ &- \mathbb{E}_{\rho_{\pi_\theta}, \rho_{\pi_+}}[\log(D_c(s,a))] - \mathbb{E}_{\rho_{\pi_-}}[\log(1 - D_c(s,a))]. \end{aligned} \tag{14}$$

We first consider the optimization problem with respect to discriminators given a fixed policy π_θ.

Proposition 3.1. *Given a fixed π_θ, maximizing the $J(\pi_\theta, D_a, D_c)$ yields to the following optimal discriminators D_a^* and D_c^*:*

$$\begin{aligned} D_a^*(s,a) &= \frac{\rho_{\pi_+(s,a)}}{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta}(s,a)}, \\ D_c^*(s,a) &= \frac{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta(s,a)}}{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta(s,a)} + \rho_{\pi_-(s,a)}}. \end{aligned} \tag{15}$$

Proof. The objective function in Eq. (14) can be rewritten as follows:

$$\begin{aligned} J_{\pi_\theta, D_a, D_c} = \int_{s,a} & [\rho_{\pi_+} \log(D_a(s,a)) + \rho_{\pi_\theta} \log(1 - D_a(s,a)) \\ &- (\rho_{\pi_+} + \rho_{\pi_\theta}) \log(D_c(s,a)) \\ &- \rho_{\pi_-} \log(1 - D_c(s,a))] dsda \end{aligned} \tag{16}$$

Considering the function inside the integral, given s, a, we maximize this function w.r.t D_a, D_c to find D_a^* and D_c^*. We can obtain the following results by setting the derivatives w.r.t D_a and D_c to 0,

$$\begin{aligned} \frac{\rho_{\pi_+(s,a)}}{D_a} - \left(\rho_{\pi_+(s,a)} + \rho_{\pi_\theta(s,a)}\right) &= 0, \\ \frac{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta(s,a)}}{D_c} - \left(\rho_{\pi_+(s,a)} + \rho_{\pi_\theta(s,a)} + \rho_{\pi_-(s,a)}\right) &= 0. \end{aligned} \tag{17}$$

Thus it's easy to verify that, $D_a^*(s,a) = \frac{\rho_{\pi_+(s,a)}}{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta}(s,a)}$ and $D_c^*(s,a) = \frac{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta(s,a)}}{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta(s,a)} + \rho_{\pi_-(s,a)}}$. Additionally, The second derivations: $-\frac{\rho_{\pi_+(s,a)}}{D_a^2}$ and $-\frac{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta(s,a)}}{D_c^2}$ are non-positive, thus verifying that we have obtained the maximum solution, concluding the proof. □

With the definition of occupancy measure $\rho_\pi(s,a)$, which indicates distribution of state-action pairs π interacts with the environment, we will show that ACIL finds a policy whose occupancy measure minimizes the JS divergence to the occupancy measure of positive trajectories ρ_{π_+}, which maximize the JS divergence to the occupancy measure of negative trajectories ρ_{π_-}.

Proposition 3.2. *Given D_a^* and D_C^*, the objective of ACIL is equivalent to minimize the following new imitation learning algorithm,*

$$\min_{\pi_\theta} D_{JS}(\rho_{\pi_+} \| \rho_{\pi_\theta}) - D_{JS}(\rho_{\pi_+} + \rho_{\pi_\theta} \| \rho_{\pi_-}) - \lambda H(\pi_\theta) \tag{18}$$

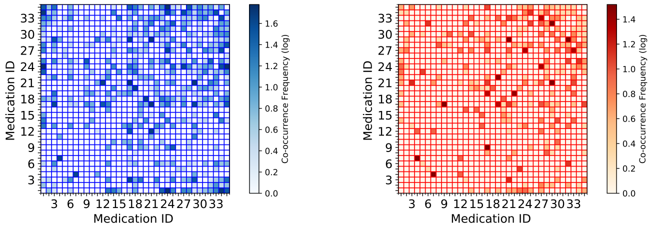


Figure 3: Comorbidity medication co-occurrence distribution of survived patients (blue) and deceased patients (red)

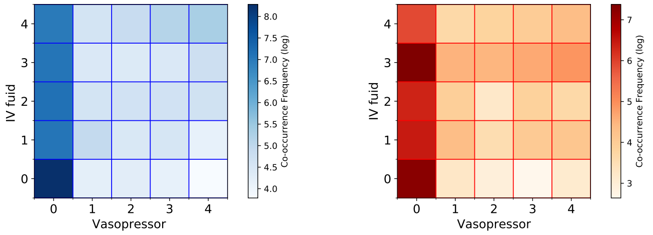


Figure 4: Sepsis medication co-occurrence distribution of survived patients (blue) and deceased patients (red)

Table 2: Dataset statistics

Dataset	#Survived	#Deceased	#Medication	#Disease
Comorbidity	16,508	3,685	35	2,000
Sepsis	6,620	3,569	25	1

Proof. Substituting D_a^* and D_c^* into the objective function defined in Eq. (14), we have,

$$\begin{aligned} J_{\pi_\theta, D_a^*, D_c^*} = & \mathbb{E}_{\rho_{\pi_+}} \left[\log \frac{\rho_{\pi_+}(s,a)}{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta}(s,a)} \right] \\ & + \mathbb{E}_{\rho_{\pi_\theta}} \left[\log(\frac{\rho_{\pi_\theta}(s,a)}{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta}(s,a)}) \right] \\ & - \mathbb{E}_{\rho_{\pi_+}, \rho_{\pi_\theta}} \left[\log(\frac{\rho_{\pi_+}(s,a) + \rho_{\pi_\theta}(s,a)}{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta}(s,a) + \rho_{\pi_-}(s,a)}) \right] \\ & - \mathbb{E}_{\rho_{\pi_-}} \left[\log(\frac{\rho_{\pi_-}(s,a)}{\rho_{\pi_+(s,a)} + \rho_{\pi_\theta}(s,a) + \rho_{\pi_-}(s,a)}) \right] \\ \propto & D_{JS}(\rho_{\pi_+} \| \rho_{\pi_\theta}) - D_{JS}(\rho_{\pi_+} + \rho_{\pi_\theta} \| \rho_{\pi_-}), \end{aligned} \tag{19}$$

which minimizes the JS divergence between occupancy measures that encourages the trajectories generated from π_θ recover the positive trajectories while differ from the negative ones. This concludes the proof. □

4 EXPERIMENTS

In this section, we conduct experiments to evaluate the proposed ACIL model. We first describe the dataset and comparison methods, followed by quantitative and qualitative studies.

4.1 Dataset Description

The experiments are conducted on a public EHRs dataset MIMIC-III [19], which contains 43K patients in critical care units during 2001 and 2012. There are 6,695 distinct diseases and 4,127 drugs in MIMIC-III. The median number of diseases of each record is 9 (Q1-Q3: 6-15). We extracted the *Comorbidity* patients following the procedure in [2], we extract the top 35 most medications and top 2,000 most diseases (ICD-9 codes) which cover 85.4% of all medication records and 95.3% of all diagnosis records, respectively. The co-occurrence distribution of the top 35 medication is shown in Fig. 3. We extract *Sepsis* patients conforming to the Sepsis-3 criteria [32]. We defined a 5×5 action space for the medical interventions covering the space of intravenous (IV) fluid and maximum vasopressor (dosage in a given 4 hour window). The action space was restricted to these two interventions as both drugs are extremely important in the management of septic patients. Sepsis medication co-occurrence distribution is depicted in Fig. 4. Additionally, the statistics of the extracted datasets used in this paper are summarized in Table 2.

For each patient, we extract relevant physiological parameters with the suggestion of clinicians, which include static variables and time-series variables. Finally, we select 8 demographic features which are static and 12 clinical variables shown in Table 3. These variables are first rescaled to z-scores, then rescaled to [0, 1]. We impute the missing variable with k-nearest neighbors and remove admissions with more than 10 missing variables. Each hospital admission of a patient is regarded as a treatment plan. Time-series data in each treatment plan is divided into different units, each of which is set to 24 hours since it is the median of the prescription frequency in MIMIC-III. If several data points are in one unit, we use their average values instead. We remove patients less than 18 years old because of the special conditions of minors. Finally, we obtain 20,193 hospital admissions of comorbidity patients (16,508 survived patients and 3,685 deceased patients) and 10,189 hospital admissions of Sepsis patients (6,620 survived patients and 3,569 deceased patients). We randomly divide the two datasets for training, validation, and testing sets by the proportion of 80%/10%/10%.

4.2 Metrics and Baselines

Canonical metrics for multi-label learning task are adopted to measure the degree of consistency between recommended prescriptions and those from doctors' prescriptions, including macro and micro average of the AUC scores (denoted as MA-AUC and MI-AUC), and Jaccard coefficient. For a patient a_n, let U_n^* be the medication set given by doctors and U^n be the medication set recommended from learned policies. The mean Jaccard is defined as

$$\frac{1}{N} \sum_{n=1}^{N} \frac{| U_n \cap U_n^* |}{| U_n \cup U_n^* |}, \tag{20}$$

where N is the number of patients. Note that the above three metrics are calculated on the positive trajectories because our goal is to recover the positive behavior.

Additionally, mortality rate is also been estimated with off-policy policy evaluation, which utilizes a set of previously-collected trajectories to estimate the value of the learned policy (π_θ) without interacting with the environment [10, 26]. In this paper we use

Table 3: Description of demographics, clinical variables and medications

Demographics	gender, age, weight, height, religion, language, marital status, and ethnicity
Lab Tests & Vital Signs	diastolic blood pressure, Glasgow coma scale, systolic blood pressure, fraction of inspired O2, heart rate, pH, respiratory rate, blood glucose, body temperature, blood oxygen saturation, blood glucose, and urine output
Medications	**Comorbidity**: Ondansetron, Quinapril, Paroxetine, Azithromycin, Guaiacolsulfonate Bisacodyl, Ondansetron, Dextromethorphan-K, Lorazepam, Acetaminophen, Metoprolol, Oxazepam, Tizanidine, Etomidate, Sirolimus, Duloxetine, Clonazepam, Fluoxetine, Alteplase, Bumetanide, Dobutamine, Simethicone, DopAmine, Carvedilol, Linezolid, Vasopressin, Azithromycin, Nephrocaps, Spironolactone, Guaifenesin, Allopurinol, Erythromycin, Dexmedetomidine, Metformin, Pravastatin **Sepsis**: five dosages of IV fluid, five dosages of Vasopressor.

the Doubly Robust Off-policy Value Evaluation [16] to obtain an unbiased estimation of the value of the learned policies.

The baselines compared in this paper are described as follows:

- **Behavior Cloning (BC)**: It cuts the trajectories into state-action tuples and learns a policy from the demonstrations via supervised learning.
- **SRL-RNN [35]**: It manually designs a sparse reward function that assigns $r_T = 15$ if a patient discharges in the end, and $r_T = -15$ if the patient dies; $r_t = 0$ when $t = 0, 1, \ldots, T-1$.
- **D3Q [28]**: This method designed a reward function as SRL-RNN and trains the policy via deep Q-learning.
- **GAIL$^+$ and GAIL$^{+,-}$ [15]**: GAIL utilizes GAN to solve the imitation learning problems which learns a policy directly by approaching the expert trajectories via the reward signals provided by the discriminator. GAIL$^+$ only takes the positive trajectories while GAIL$^{+,-}$ uses both positive and negative trajectories.

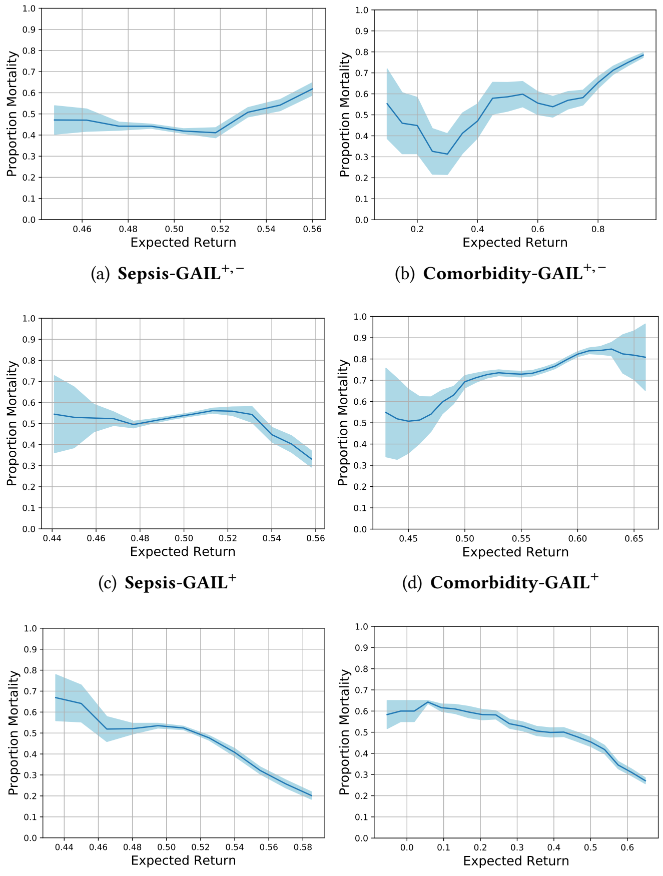
To ensure fair comparisons, we use the same neural network architecture for all baselines. The discriminator networks of GAIL and ACIL used the same 4-layer MLPs as the policy network with an embedding size of 64. The weights of the two discriminators in ACIL are both set as 0.5. We utilize Adam to optimize all the models.

4.3 Results

4.3.1 Model Comparisons. Table 4 summarizes the performance of the baselines evaluated by the four metrics. We observe that: 1) Behavior cloning has higher mortality rate over the other baselines. The reason is that, with the i.i.d assumption on states, behavior

Table 4: Model comparisons on Comorbidity and Sepsis

Methods	Comorbidity			
	Mortality Rate	MA-AUC	MI-AUC	Jaccard
BC	0.265	0.614	0.688	0.375
SRL-RNN	0.234	0.625	0.699	0.387
D3Q	0.238	0.531	0.609	0.219
GAIL$^{+,-}$	0.252	0.601	0.679	0.366
GAIL$^+$	0.240	0.623	0.692	0.382
ACIL	**0.214**	**0.646**	**0.715**	**0.418**
	Sepsis			
BC	0.401	0.541	0.646	0.327
SRL-RNN	0.379	0.539	0.640	0.339
D3Q	0.394	0.532	0.641	0.326
GAIL$^{+,-}$	0.393	0.537	0.639	0.334
GAIL$^+$	0.381	0.545	0.653	0.349
ACIL	**0.369**	**0.578**	**0.671**	**0.372**

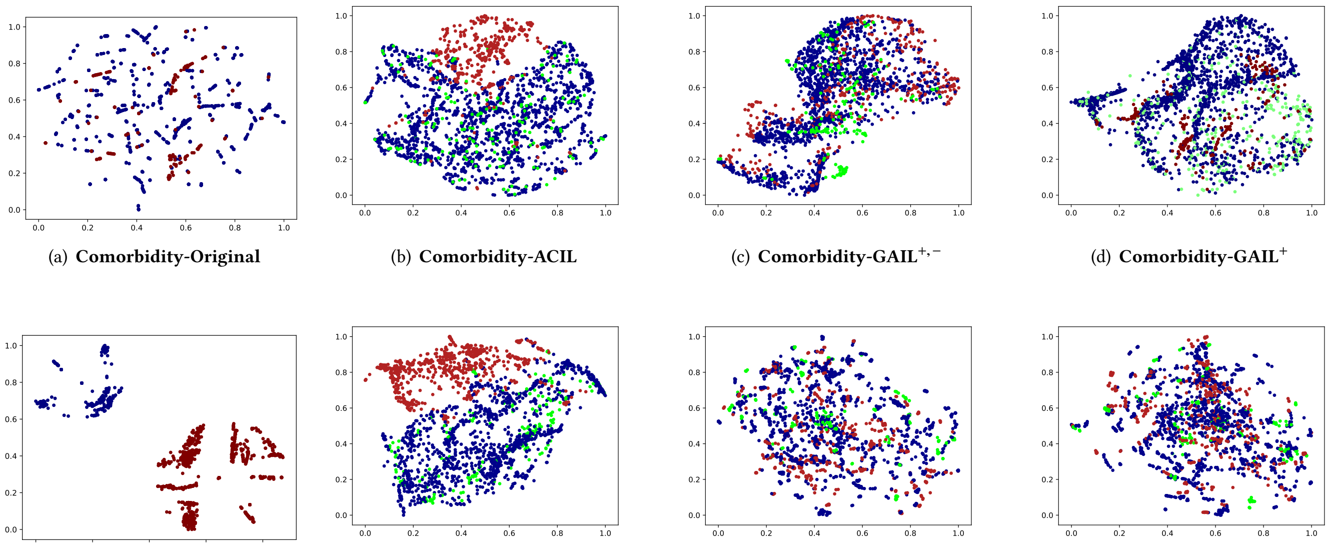


(a) **Sepsis-GAIL$^{+,-}$** (b) **Comorbidity-GAIL$^{+,-}$**

(c) **Sepsis-GAIL$^+$** (d) **Comorbidity-GAIL$^+$**

(e) **Sepsis-ACIL** (f) **Comorbidity-ACIL**

Figure 5: Mortality vs. expected return curve computed by the learned policies of different models with 2,000 Bootstrap samples. The shaded area represents the standard error of mean.



(a) **Comorbidity-Original**

(b) **Comorbidity-ACIL**

(c) **Comorbidity-GAIL$^{+,-}$**

(d) **Comorbidity-GAIL^{+}**

(e) **Sepsis-Original**

(f) **Sepsis-ACIL**

(g) **Sepsis-GAIL$^{+,-}$**

(h) **Sepsis-GAIL^{+}**

Figure 6: Visualization of the patients embedding. Green nodes indicate the patient embedding generated by different policies. Red nodes are the deceased patients' embedding and blue nodes are the survived patients' embedding.

cloning suffers from compounding errors as policy unrolled. 2) With the mixture of positive and negative trajectories, adversarial imitation learning GAIL$^{+,-}$ follows the sub-optimal policies and results in a worse performance compared to GAIL^{+}. 3) Reinforcement learning algorithm (SRL-RNN and D3Q) is not effective with the sparse reward functions. Imitation learning can alleviate this problem by obtaining the reward signal from demonstrations. 4) ACIL consistently outperforms all baselines. It's because ACIL considers discovering DTRs as a sequential decision making problem and focuses on the long-term influence on the current action. Additionally, with the usage of both positive and negative demonstrations, ACIL is able to mimic the positive policies while avoiding the mistakes (negative demonstrations).

4.3.2 Mortality vs. Expected Return. We consider the output of the discriminators of GAIL^{+}, GAIL$^{+,-}$ and ACIL as the reward. The expected returns of each patient in the test datasets can be calculated with this reward signal. The relation between expected returns and mortality rates is shown in Figure 5. It can be seen from the figure that ACIL has a more clear negative correlation between expected returns and mortality rates, as indicated in Fig 5 (e)-(f), than the other adversarial methods, which demonstrates that ACIL can well evaluate the value of the policy.

4.4 Visualization

4.4.1 Trajectories Embedding. We extract the first-layer hidden variables of the policy network as the embedding of the patients with the input of the states. The patients' representations learned by different methods are visualized with t-SNE [23] as shown in Fig. 6. Each node indicates the states of a patient: blue nodes present the survived patients, red nodes indicate the deceased patients, green nodes are the generated patients with the treatments given by the different methods. We obtain an embedding of the patients with the average of the sequence states of the patients at each time step. We observe that: (1) The original state distribution which is visualized by the original states of the patients shows that the states of survived patients and deceased patients are hard to distinguish in Comorbidity dataset, while they are easy to differentiate among sepsis patients. (2) The embedding learned by ACIL can correctly distinguish the negative trajectories and positive trajectories, which provides empirical evidences for the effectiveness of our methods. It also demonstrates that, leveraging the negative samples can help learn a policy that mimics the positive demonstrations while differs from the demonstrations. (3) GAIL^{+} and GAIL$^{+,-}$ are easy to generated similar trajectories as the negative policies, which are not effective.

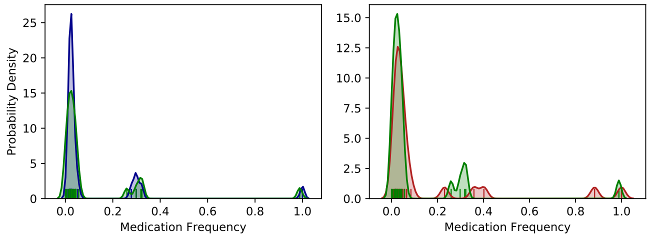


Figure 7: Action frequency comparison on Sepsis dataset among positive treatments (blue), negative treatments (red) and ACIL's policy (green).

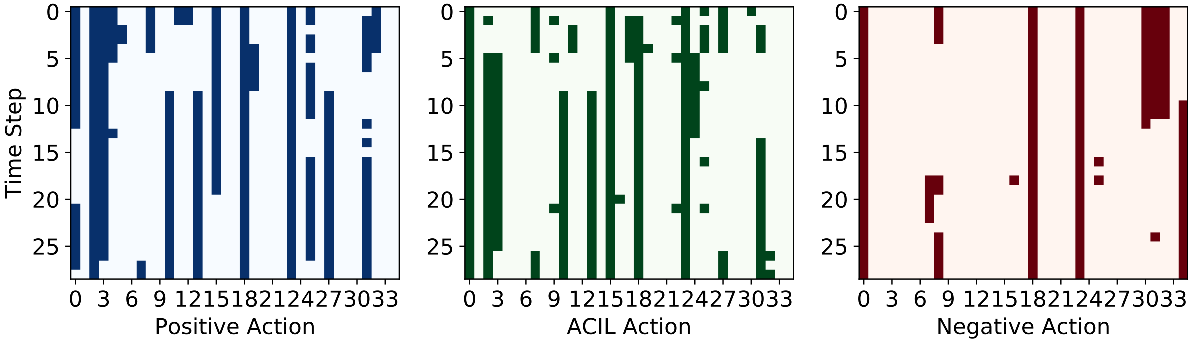


Figure 8: Prescriptions for patients in Comorbidity

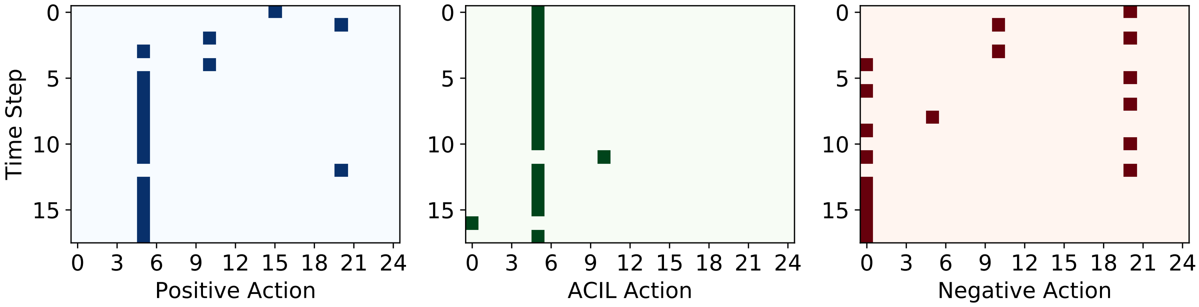


Figure 9: Prescriptions for patients in Sepsis

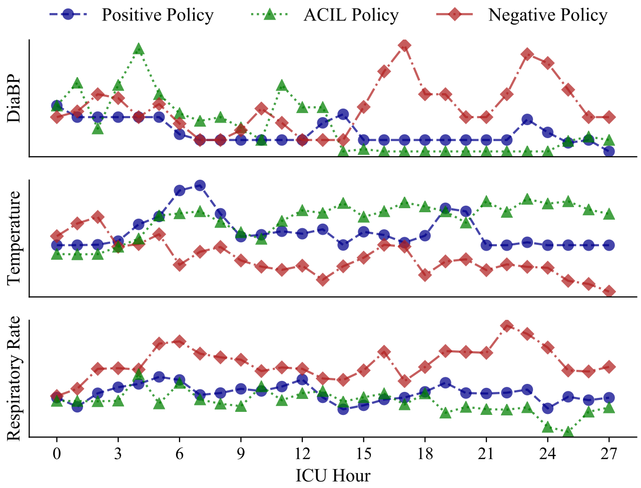


Figure 10: States of the patients in Comorbidity

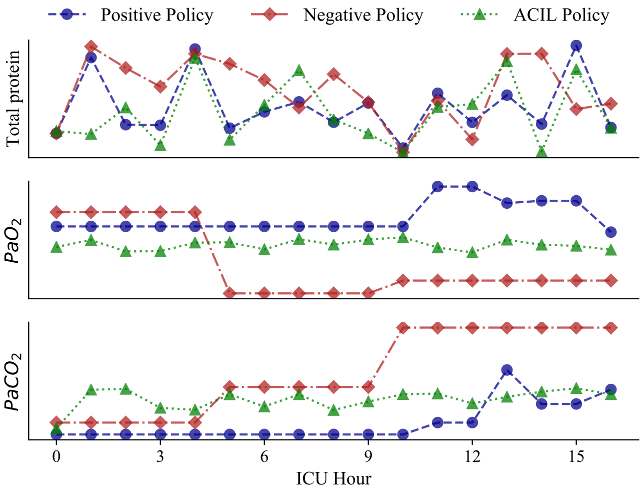


Figure 11: States of the patients in Sepsis

4.4.2 Policy Distribution Analysis. In the dataset description Section 4.1 we show that there are some differences between the distributions derived from the positive demonstrations and negative demonstrations (as shown in Fig. 3-4). In this part, we count the action frequency of the positive policy, negative policy and ACIL's policy, and plot their distributions in Fig. 7. Notably, the policy learned by ACIL is close to the positive demonstrations while different from the negative demonstrations, which verified the effectiveness of the negative demonstrations for learning a policy.

4.5 Case Study

We select two patients from Comorbidity dataset (with lung cancer) and two patients from Sepsis dataset with similar initial states (i.e., similar ages and lab test results), as shown in Fig. 10 and 11. The patients with blue prescription (positive actions) were survived and the patients with red prescription (negative actions) were deceased. We sample two initial state that are similar to these four patients, and apply ACIL with the two initial states to generate the state trajectories (marked in green) on the patient model, and all the patients with green trajectories are survived finally. It shows that ACIL can learn a policy which is close to the positive actions while stays far away from the negative actions, as shown in Fig. 8 and 9. The x-axis in the figure indicates the action index and the y-axis is the ICU hour of the patient. Each element in i^{th} row and j^{th} column with colors indicates the j-th medication is given for the patient at time step i. Similarly, the state trajectories generated by ACIL can also match the trajectories generated by the positive policies. This result demonstrates that both the positive and negative trajectories can help ACIL learn a policy. The positive demonstrations teach ACIL to learn what to do, and the negative demonstrations teach ACIL to learn what not to do.

4.6 Model Analysis

4.6.1 Convergence Analysis. Figure 12 presents the loss of the two discriminators and the return of the learned policy obtained in each learning epoch. Notably, ACIL is able to stably converge which is coordinated with our analysis.

4.6.2 Parameter Sensitivity. Figure 13 shows the effectiveness of the weight parameter ω_α, which is used to balance adversarial discriminator and cooperative discriminator. It can be inferred from the figure that, when taking value $\geq$ 0.5, the model can achieve relatively high Jaccard, MI-AUC and MA-AUC and a lower mortality rate.

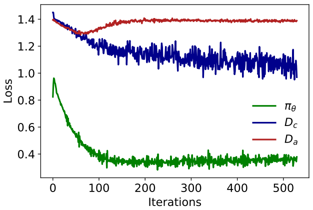


Figure 12: Convergence analysis

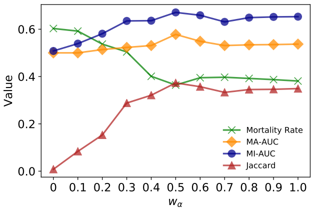


Figure 13: Parameter analysis on ω_α

5 RELATED WORK

5.1 Imitation Learning

Imitation learning, also known as *learning from demonstrations*, can be generally grouped into three categories: behavior cloning (BC), inverse reinforcement learning (IRL) and adversarial imitation learning (AIL). BC [25] is a form of supervised learning which is to learn a direct mapping from the states to the actions. BC can avoid interacting with the environment. However, without substantial correction during training, BC is known to have compounding error [30].

Instead of directly utilizing the supervised signal from demonstrations, inverse reinforcement learning [1, 39] finds a reward function that models the intention of the demonstrator. The learned reward function gives feedback to the states that were un-visited. A policy can be learned by reinforcement learning methods [33] with this reward function. Maximum entropy IRL [11, 36, 40] seeks to find a reward function that makes the demonstrations achieve highest total reward as well as maximize the entropy of the resultant policy. Though these methods can alleviate the compounding error issue, they learn a policy indirectly with an inner loop of reinforcement learning, which is costly.

Adversarial imitation learning [15] leverages generative adversarial networks [12] to directly learn the policy and reward function simultaneously, where the policy corresponds to the generator and the discriminator plays as the reward function.

Most of the imitation learning methods work with success demonstrations. However, the agent can also be trained from the failure demonstrations to learn what not to do [13]. Here in ACIL, it incorporates both success and failure demonstration trajectories.

5.2 Treatment Recommendation

BC [2, 7, 17, 37] and reinforcement learning [3, 21, 28, 31, 35, 37] are two major methods used to learn DTRs. When the EHRs are plentiful and optimal, BC can effectively recover the doctor's policies. However, due to the dynamics of the treatment process, BC methods are easy to introduce the compounding error. The inefficiencies of BC come from the sequential nature of this problem. In BC, even a slight errors in mimicking the demonstration behavior can quickly accumulate as the policy unrolled [9, 29]. To correct the mistakes, the corrective behaviors should appear frequently. In addition, The failure samples are usually discarded, either explicitly by researchers, or implicitly in the algorithms themselves in BC, which reduce the sample efficiency. (The EHR datasets are very limited, we should make full use of them.)

To make use of both success and failure datasets, RL based methods can directly learn a policy via the goal of maximizing the long-term reward of patients [3, 21, 28, 35]. However, they requires hand-crafted, knowledge of the true signs of the rewards features. In addition, the learned policy is highly rely on the accuracy of the pre-defined reward function.

6 CONCLUSIONS

In this paper, we propose ACIL to learn the optimal dynamic treatment regimes with both positive and negative demonstration trajectories as inputs. The learned policy is able to mimics the positive demonstrations while differs from the negative demonstrations with two discriminators: an adversarial discriminator is used to minimize the discrepancies between the demonstrations generated from the policy and the positive demonstrations, and a cooperative discriminator is used to distinguish the negative demonstrations from the positive and generated demonstrations. ACIL utilizes the reward signals from the discriminators to refine the policy and the patient model built with variational autoencoders. Empirical results on MIMIC-III demonstrate that ACIL improves the likelihood of patient survival and provides better dynamic treatment regimes with the usage of all treatment demonstrations.

ACKNOWLEDGMENTS

The work is partially supported by NSF DGE-1829071, and the National Key Research and Development Program of China under

Grant No. 2016YFB1000904. We thank the anonymous reviewers for their careful reading and insightful comments on our manuscript.

REFERENCES

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

[2] Jacek M Bajor and Thomas A Lasko. 2016. Predicting medications from diagnostic codes with recurrent neural networks. (2016).

[3] Melanie K Bothe, Luke Dickens, Katrin Reichel, Arn Tellmann, Bjoern Ellger, Martin Westphal, and Ahmed A Faisal. 2013. The use of reinforcement learning algorithms to meet the challenges of an artificial pancreas. *Expert review of medical devices* 10, 5 (2013), 661–673.

[4] Lars Buesing, Theophane Weber, Sebastien Racaniere, SM Eslami, Danilo Rezende, David P Reichert, Fabio Viola, Frederic Besse, Karol Gregor, Demis Hassabis, et al. 2018. Learning and querying fast generative models for reinforcement learning. *arXiv preprint arXiv:1802.03006* (2018).

[5] Bibhas Chakraborty and Susan A Murphy. 2014. Dynamic treatment regimes. *Annual review of statistics and its application* 1 (2014), 447–464.

[6] Xinshi Chen, Shuang Li, Hui Li, Shaohua Jiang, Yuan Qi, and Le Song. 2019. Generative Adversarial User Model for Reinforcement Learning Based Recommendation System. In *International Conference on Machine Learning*. 1052–1061.

[7] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*. 301–318.

[8] John Co-Reyes, YuXuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. 2018. Self-Consistent Trajectory Autoencoder: Hierarchical Reinforcement Learning with Trajectory Embeddings. In *International Conference on Machine Learning*. 1008–1017.

[9] Yan Duan, Marcin Andrychowicz, Bradly Stadie, OpenAI Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. 2017. One-shot imitation learning. In *Advances in neural information processing systems*. 1087–1098.

[10] Miroslav Dudík, John Langford, and Lihong Li. 2011. Doubly Robust Policy Evaluation and Learning. In *ICML*. 1097–1104.

[11] Chelsea Finn, Sergey Levine, and Pieter Abbeel. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In *International Conference on Machine Learning*. 49–58.

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*. 2672–2680.

[13] Daniel H Grollman and Aude Billard. 2011. Donut as I do: Learning from failed demonstrations. In *2011 IEEE International Conference on Robotics and Automation*. IEEE, 3804–3809.

[14] Shixiang Gu, Ethan Holly, Timothy Lillicrap, and Sergey Levine. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *ICRA*. IEEE, 3389–3396.

[15] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. In *Advances in neural information processing systems*. 4565–4573.

[16] Nan Jiang and Lihong Li. 2015. Doubly robust off-policy value evaluation for reinforcement learning. *arXiv preprint arXiv:1511.03722* (2015).

[17] Bo Jin, Haoyu Yang, Leilei Sun, Chuanren Liu, Yue Qu, and Jianing Tong. 2018. A treatment engine by predicting next-period prescriptions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 1608–1616.

[18] Junqi Jin, Chengru Song, Han Li, Kun Gai, Jun Wang, and Weinan Zhang. 2018. Real-time bidding with multi-agent reinforcement learning in display advertising. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*. ACM, 2193–2201.

[19] 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. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.

[20] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[21] Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. 2018. The Artificial Intelligence Clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine* 24, 11 (2018), 1716.

[22] Yunzhu Li, Jiaming Song, and Stefano Ermon. 2017. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*. 3812–3822.

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

[24] Susan A Murphy. 2003. Optimal dynamic treatment regimes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65, 2 (2003), 331–355.

[25] Dean A Pomerleau. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation* 3, 1 (1991), 88–97.

[26] Doina Precup, Richard S. Sutton, and Sanjoy Dasgupta. 2001. Off-policy temporal-difference learning with function approximation. In *ICML*. 417–424.

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

[28] Aniruddh Raghu, Matthieu Komorowski, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. 2017. Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422* (2017).

[29] Stéphane Ross and Drew Bagnell. 2010. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. 661–668.

[30] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. 627–635.

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

[32] 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. 2016. The third international consensus definitions for sepsis and septic shock (Sepsis-3). *Jama* 315, 8 (2016), 801–810.

[33] Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning*. Vol. 2. MIT press Cambridge.

[34] Elise Van der Pol and Frans A Oliehoek. 2016. Coordinated deep reinforcement learners for traffic light control. *Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016)* (2016).

[35] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. 2018. 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*. ACM, 2447–2456.

[36] Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. 2015. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888* (2015).

[37] Yutao Zhang, Robert Chen, Jie Tang, Walter F Stewart, and Jimeng Sun. 2017. LEAP: learning to prescribe effective and safe treatment combinations for multimorbidity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1315–1324.

[38] Guanjie Zheng, Fuzheng Zhang, Zihan Zheng, Yang Xiang, Nicholas Jing Yuan, Xing Xie, and Zhenhui Li. 2018. DRN: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 167–176.

[39] Shao Zhifei and Er Meng Joo. 2012. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics* 5, 3 (2012), 293–311.

[40] Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning. (2008).