Extracting Clinician's Goals by What-if Interpretable Modeling

Chun-Hao Chang [1 2 3] **George Alexandru Adam** [1 2 3] **Rich Caruana** [4] **Anna Goldenberg** [1 2 3]

Abstract

Although reinforcement learning (RL) has tremendous success in many fields, applying RL to real-world settings such as healthcare is challenging when the reward is hard to specify and no exploration is allowed. In this work, we focus on recovering clinicians' rewards in treating patients. We incorporate the what-if reasoning to explain the clinician's actions based on future outcomes. We use generalized additive models (GAMs) - a class of accurate, interpretable models - to recover the reward. In both simulation and a real-world hospital dataset, we show our model outperforms baselines. Finally, our model's explanations match several clinical guidelines when treating patients while we found the previously-used linear model often contradicts them.

1. Introduction

Reinforcement learning has achieved tremendous success in many fields including Go (Silver et al., 2017), autonomous driving (Sallab et al., 2017), and healthcare (Chang et al., 2019; Fatemi et al., 2021). However, designing reward functions for real-world problems is challenging since it is unclear how much more preferable some states are than others. For example, clinicians often administer vasopressors to increase blood pressure, but a too-high dose might cause vasopressor-induced shock. Also, when robots are designed to navigate to a specific location, the reward function has to prefer not to break nearby items or hurt the people along the path (Amodei et al., 2016). Specifying all possible conditions in the reward is very hard, and also the magnitude of the reward is not easy to determine when multiple goals are needed (e.g. treating patients while reducing the side effects).

One way to avoid reward function tuning is to do imitation learning that directly mimics what experts do by their demonstrations. However, we only extract the rule of how experts make decisions (e.g. administer vasopressors when blood pressure is low) but not the reason why they do (e.g. maintain patient's blood pressure above 65). Therefore, the rules extracted are not suitable for transferring when environments change or different actions are available, while goals recovered from inverse reinforcement learning (IRL) are more robust and allow the user to inspect and confirm if these are intended consequences.

In many settings such as medicine, users often behave based on the potential future outcomes: given the current information, what desirable future outcomes would happen if I take certain actions? For example, doctors treat patients with vasopressors to increase their blood pressure in the future. Unlike other IRL methods which use the history to explain the experts' behaviors (e.g. the doctors give vasopressors because the patient's blood pressure is dropping), we instead uses the potential future outcome of the patients (e.g. the doctors want to maintain the blood pressure above 65 in the next few hours). We believe it's more closer to what doctors think. Importantly, the learned preference is transferable across different environments when actions are different (e.g. different hospitals have different treatment protocols). Finally, it is of interest to clinicians to understand if their behavior matches the intended goals, and helps serve as a sanity check tool when designing the reward. Besides the clinical setting, we think it's useful for discovering goals for agents who can not explain themselves (e.g. animals).

Generalized Additive Models (GAMs) have been in popular use since the 80s serving as important tools to understand dataset patterns in many fields including healthcare, business and science (Chang et al., 2021b). GAMs are also used to audit black-box models (Tan et al., 2018) or discover fairness bias in the data (Tan et al., 2019). As a white-box model by design, it is surprisingly effective compared to black-box counterparts such as deep neural networks for tabular data. However, to the best of our knowledge, it has not been used in IRL to understand experts' goals.

In this work, we first recover the potential future outcomes from an observational data by counterfactual modeling in which we make some causal assumptions to identify the effects. Then we use the learned future outcomes to recover the clinicians' rewards by an interpretable GAM model in an

[1]University of Toronto [2]Vector Institute [3]Hospital of Sick-kids [4]Microsoft Research. Correspondence to: Chun-Hao Chang <chkchang21@gmail.com>.

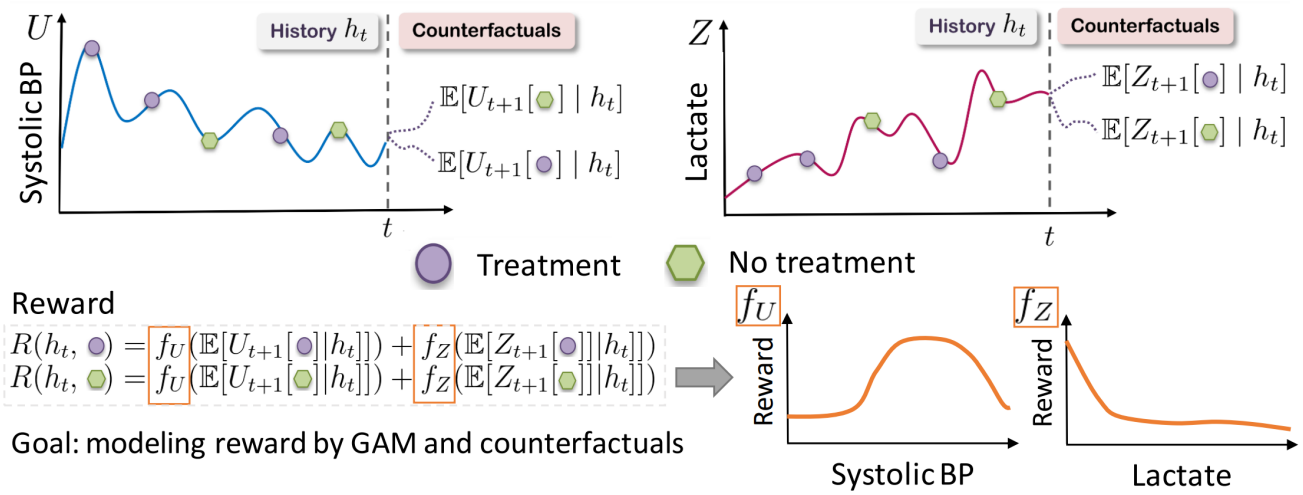

Figure 1. The overview of our work. We first learn a model that predicts future counterfactuals. Then we recover the reward of clinicians by a GAM model based on the estimated counterfactuals. Finally, we interpret what doctors' rewards are from the GAM graphs.

Adversarial IRL (AIRL) framework, and thus call our model Counterfactual AIRL (CAIRL). In our sepsis simulation, we show CAIRL outperforms both AIRL (Qureshi et al., 2018) and state-of-the-art Counterfactual IRL (CIRL) (Bica et al., 2020) by having a higher reward and recovering the underlying goals better. In a real-world clinical management task (hypotension), our GAM model recovers meaningful clinical thresholds and patterns while the previously-used linear model often contradicts them.

2. Related Work

We summarize prior works in Table 1. Several methods have been proposed to recover the reward function based on expert demonstrations. Max-margin Apprenticeship Learning (MMA, Abbeel & Ng (2004)) assumed the existence of an expert policy π_E that is optimal under some unknown *linear* reward function of the form $R(s,a) = \boldsymbol{w} \cdot \phi(s,a)$ for some reward weights $\boldsymbol{w} \in R^d$ and the feature map $\phi(s,a)$. However, to evaluate how well a policy behaves, they require environmental dynamics to be known. LSTD-Q (Klein et al., 2011) relaxes it by learning to evaluate policy performance via temporal difference method that resembles Q-learning. DFSN (Lee et al., 2019) further improves upon LSTD-Q by using a neural net and prioritized experience replay (Schaul et al., 2015). However, they can only be used to evaluate policies similar to the expert policy. CIRL (Bica et al., 2020) instead learns a counterfactual transition model, and models expert rewards on the estimated future states instead of current states, achieving the state-of-the-art performance. Unfortunately, these MMA methods do not have a unique solution and may not guarantee to recover the expert's reward, since even $\boldsymbol{w} = \mathbf{0}$ is a solution to their optimization (Ziebart et al., 2008). Additionally, the linear assumption in these works is too restrictive for many real-world problems including healthcare, where the goal usually is to maintain patients' vitals in a middle range (e.g. temperature between 36-38) but the linear model only allows monotonically increasing or decreasing relationships.

To solve the non-unique solution problem, Max-Ent IRL (Ziebart et al., 2008) proposes Max-Entropy IRL that seeks to find a reward r that maximizes the likelihood of the trajectories under the optimal policy π_E. This formulation has a unique global solution and solves the trivial solution problem in MMA, but still assumes the reward is linear. GAIL (Ho & Ermon, 2016) is the first work to formulate Max-Ent IRL as an adversarial game between a policy learner (generator) and a reward model (discriminator) and thus allows non-linear reward modeled by a deep neural network. They prove that their learned policy recovers the expert policy, but the learned reward may not recover the expert reward. AIRL (Qureshi et al., 2018) further improves upon GAIL by avoiding the degradation of the reward model (discriminator) in GAIL, and presents a practical implementation that scales well to various environments. Although AIRL recovers the reward, the adoption of DNNs in the reward modeling hinders the interpretability to understand the expert reward. They also did not consider what-if reasoning and the difficult batch setting in this paper.

iAIRL (Srinivasan & Doshi-Velez, 2020) also aims to recover the clinician's reward and uses an interpretable differential decision tree (DNDT) (Yang et al., 2018) following the AIRL framework. However, due to the exponential feature combinations of DNDT, iAIRL only modeled 5 features. Their performance is much lower compared to a deep neural network (64% v.s. 71%) while using GAMs results in similar state-of-the-art performance and achieving

Table 1. The comparison of related works.

Method	Batch data	What-if reasoning	Reward	Have unique solution	Interpretable	Modeled more than 5 features
MMA (Abbeel & Ng, 2004)	✗	✗	$w \cdot \phi(s)$	✗	✗	✓
DSFN (Lee et al., 2019)	✗	✗	$w \cdot \phi(s,a)$	✗	✗	✓
CIRL (Bica et al., 2020)	✓	✓	$w \cdot \mathbb{E}[Y_{t+1}\|h]$	✗	✗	✓
AIRL (Qureshi et al., 2018)	✗	✗	DNN(s)	✓	✗	✓
iAIRL (Srinivasan & Doshi-Velez, 2020)	✓	✗	DNDT(s)	✓	✓	✗
CAIRL (ours)	✓	✓	GAM($\mathbb{E}[Y_{t+1}\|h]$)	✓	✓	✓

interpretability. Besides, they did not consider the what-if reasoning which improves performance in our experiment.

3. Background

Markov Decision Process (MDP) We adopt the standard notations of MDP. An MDP consists of a tuple $(S, A, T, T_0, R, \gamma)$ where $s \in S$ states, $a \in A$ actions (discrete, in this work), $T(s'|s,a)$ the transition probabilities, and T_0 the initial state distribution, $R(s,a)$ the reward function, and $\gamma \in [0,1)$ the discount factor. A policy $\pi(a|s)$ gives the probability of taking an action a in a state s. An optimal policy π^* maximizes the cumulative reward G:

$$G_\pi = \sum_{t=0}^{T} E_{s_{t+1}\sim T(s_t,a_t), a_t\sim\pi(s_t)}[\gamma^t r(s_t,a_t)]$$
$$\pi^* = \text{argmax}_\pi G_\pi$$

In the Batch Inverse Reinforcement Learning (IRL) setting, an agent is given some trajectories (s,a) from a policy which we are told is (near) optimal, and in turn, asked to determine what the reward $R(s,a)$ must have been. Further we assume the "batch" setting which means the agent has no further interaction with the MDP, resembling high-stakes scenarios in real life such as healthcare.

Generalized Additive Models (GAM) GAMs have emerged as a leading model class that is accurate (Caruana et al., 2015), and yet simple enough for humans to understand and mentally simulate how a GAM model works (Hegselmann et al., 2020; Kaur et al., 2020), and is widely used in scientific data exploration (Hastie & Tibshirani, 1995) and model bias discovery (Tan et al., 2018).

GAM are interpretable by design due to their simple functional forms. Given an input $x \in \mathbb{R}^D$, a label y, a link function g (e.g. g is identity function in regression), main effects f_j for each feature j, GAM are expressed as:

$$g(y) = f_0 + \sum_{j=1}^{D} f_j(x_j).$$

Unlike common models (e.g. DNNs) that use all feature interactions, GAM is restricted to only use single feature, so the impact of each f_j can be visualized independently as a graph. That means, we may plot x_j on the x-axis and $f_j(x_j)$ on the y-axis. Note that a linear model is a special case of GAM. Humans can easily simulate the output of a GAM by reading f_js from the graph and adding them together.

Node-GAM We adopt Node-GAM (Chang et al., 2021a), a deep-learning version of GAM that optimizes multi-layer soft decision trees achieving the state-of-the-art performance. Unlike other GAMs like EBM (Nori et al., 2019), it allows back-propagation and thus can be used in our CAIRL framework.

4. Methods

See Fig. 2 for an overview. First, we introduce how we train our counterfactual transition model that estimates the next state given expert batch data. Then we illustrate how we recover the expert reward in an adversarial framework by training a policy (generator) and a reward model (discriminator) jointly. Finally, we interpret the reward model.

Counterfactual Transition Model To explain the expert's intent by what-if reasoning, we need to model the future outcome that defines the feature map ϕ. We adopt the potential outcomes framework (Rubin, 1978). Let $Y[a]$ be the potential outcome for treatment $a \in A$. Then we learn the feature map ϕ as the potential outcome of the next state given the action a_t and history h_t:

$$\phi = \mathbb{E}[Y_{t+1}[a_t]|h_t] \quad (1)$$

For factual action a_t assigned under expert policy, the factual observed outcome s_{t+1} is the same as the potential outcome $Y_{t+1}[a_t]|h_t$. And the potential outcomes for other actions are the counterfactual ones and allows us to understand what might happen if patients receive different treatments. To identify the potential outcomes from the expert batch data, we have the same assumptions as Schulam & Saria (2017); Lim et al. (2018) of Consistency, Positivity and No Hidden Confounders (Rosenbaum & Rubin, 1983; Robins et al., 2000) to estimate time-series counterfactual outcomes. See Supp. A for a more detailed discussion.

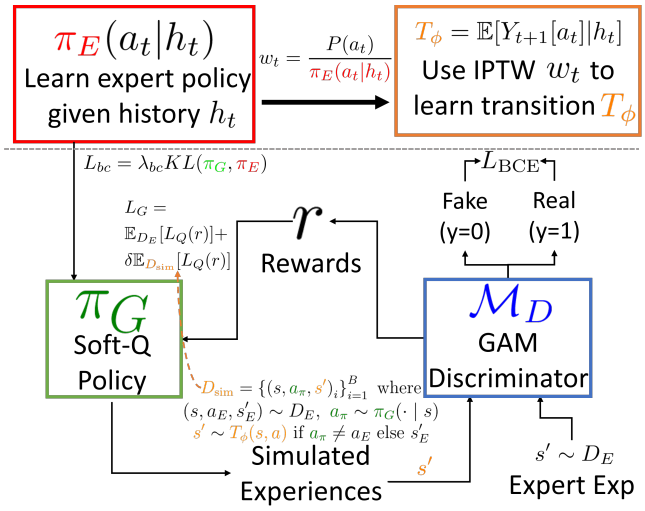

Figure 2. Our system pipeline consists of 4 models. First we train an expert policy π_E by RNN and the counterfactual transition model T_ϕ weighted by inverse propensity w_t (IPTW). Then we train the generator policy π_G and the GAM discriminator $\mathcal{M}_D$ in an adversarial way. Finally we visualize the reward learned by the GAM model $\mathcal{M}_D$.

Learning potential outcomes from expert batch data necessitates counterfactual modeling, because the expert treatments which also consider the history of patients create a time-dependent confounding bias (Schulam & Saria, 2017). We adopt the stabilized inverse propensity weighting (IPTW) framework (Robins et al., 2000; Lim et al., 2018) by adjusting the sample weights of the prediction task. Specifically, we first estimate the expert policy π_E by a recurrent neural network GRU (Gated Recurrent Unit, Chung et al. (2014)). Then in each time step t and given the marginal action probability $P(a_t)$, we set the sample weight w_t as:

$$w_t = P(a_t)/\pi_E(a_t|h_t). \tag{2}$$

We train another GRU that predicts s_{t+1} where the loss is weighted by w_t. See Supp. D for training details.

Generator: deriving optimal policy under a given reward To derive a policy under a given reward generated by discriminator, we adopt the state-of-the-art offline RL method: Soft-Q learning (Haarnoja et al., 2017) that allows optimization on both the expert batch data and the counterfactual data generated by the transition model. Specifically, given a network Q, an experience of (s, a, r, s'), and entropy coefficient α, we minimize the Huber loss $L_\mathcal{H}$:

$$\min_Q L_\mathcal{H}((Q(s,a), r(s') + \sum_{a'} \pi(a'|s')(Q(s',a') - \log \pi(a'|s')))$$

$$\text{where} \quad \pi(a|s) = \text{Softmax}(Q(s,a)/\alpha) \tag{3}$$

Since our transition model may not predict the future states perfectly, we use three ways to alleviate this bias when updating the generator. First, instead of simulating the full trajectories from our transition model acted under policy π like MMA does, we instead only do one-step future predictions from the sampled expert demonstrations as our counterfactual data; this reduces the extrapolation error in multiple timesteps. Second, when the sampled action matches the expert action in the history, we directly use the logged next state s_{t+1} instead of the prediction from the transition model. Finally, we put a smaller weight δ on the loss of these counterfactual data. We search δ over $[0, 0.5, 1]$ and find $\delta = 0.5$ produces the best result. Specifically, given the expert batch data D_E and the transition model T:

$$L = \mathbb{E}_{D_E}[L_\mathcal{H}(s,a,s')] + \delta\mathbb{E}_{s\sim D_E, a\sim\pi(\cdot|s), s'\sim T(s,a)}[L_\mathcal{H}(s,a,s')] \tag{4}$$

Behavior cloning regularization To stabilize the generator optimization, we find that it's crucial to regularize the first part of optimization to be close to the policy derived from behavior cloning (i.e. using supervised learning to predict actions). We first train a GRU model to directly predict a_t by history h_t that learns $\pi_{bc}(a_t|h_t)$; then when updating our Q-network, we add an additional KL divergence loss between the current policy π and π_{bc}:

$$L_{bc} = \lambda_{bc} KL(\pi_Q, \pi_{bc})$$

We linearly decay λ_{bc} to 0 in the first half of the training.

Discriminator: the reward model We follow the similar design from AIRL that trains a binary classifier to predict whether ϕ comes from the expert or the generator, but here ϕ is the next state s_{t+1} instead of s_t. Specifically, given g as the reward model, h as the shaping term modeling in AIRL, π the generator's policy, the discriminator logit D is:

$$D(s,a,s') = g(\phi) + h(s') - h(s) - \log \pi(s,a)$$

And we set feature map ϕ as s' while AIRL sets ϕ as s. We set both g and h as the Node-GAM. Then we set the class y of expert data as 1 and generated data as 0, and optimize the binary cross entropy loss (BCE):

$$\begin{aligned} L_D = &\mathbb{E}_{s,a,s'\sim D_E}[\text{BCE}(D(s,a,s'), \mathbf{1})] + \\ &\mathbb{E}_{s\sim D_E, a\sim\pi(\cdot|s), s'\sim T(s,a)}[\text{BCE}(D(s,a',s'), \mathbf{0})]. \end{aligned}$$

Discriminator Stabilizing Tricks When optimizing discriminator, we use both one-sided label smoothing (Salimans et al., 2016) and add a small Gaussian noise (Jenni & Favaro, 2019) to the inputs which have been shown useful to stabilize GAN adversarial optimization (see Supp. E.1).

Table 2. The performance for 7 IRL models in sepsis simulation environment. Here we show both the reward and the L1 distance (Dist) to the ground truth reward in the GAM shape graph. The mean and stdev are calculated in 5 runs. The best numbers are bolded.

	$\gamma = 0.9$				$\gamma = 0.5$			
	GAM MDP		Linear MDP		GAM MDP		Linear MDP	
	Reward	Dist	Reward	Dist	Reward	Dist	Reward	Dist
MMA	-6.112 $\pm$ 0.027	-	1.631 $\pm$ 0.004	-	-1.081 $\pm$ 0.010	-	0.316 $\pm$ 0.001	-
CIRL	-7.637 $\pm$ 0.040	-	1.629 $\pm$ 0.013	-	-1.111 $\pm$ 0.001	-	0.341 $\pm$ 0.003	-
Linear-AIRL	-	-	1.690 $\pm$ 0.011	0.051	-	-	0.343 $\pm$ 0.002	0.358
FCNN-AIRL	-0.919 $\pm$ 0.012	-	1.664 $\pm$ 0.010	-	-0.362 $\pm$ 0.003	-	**0.344** $\pm$ 0.003	-
GAM-AIRL	-0.931 $\pm$ 0.012	0.358	1.670 $\pm$ 0.013	0.210	**-0.357** $\pm$ 0.003	0.421	0.342 $\pm$ 0.003	0.234
Linear-CAIRL	-6.449 $\pm$ 0.008	0.471	**1.698** $\pm$ 0.032	**0.016**	-1.003 $\pm$ 0.012	0.547	0.343 $\pm$ 0.003	0.343
FCNN-CAIRL	-0.947 $\pm$ 0.010	-	1.687 $\pm$ 0.009	-	**-0.357** $\pm$ 0.004	-	0.343 $\pm$ 0.004	-
GAM-CAIRL	**-0.894** $\pm$ 0.013	**0.282**	1.682 $\pm$ 0.022	0.073	**-0.357** $\pm$ 0.001	**0.345**	**0.344** $\pm$ 0.004	**0.195**
Expert	-0.883 $\pm$ 0.002	0.000	1.708 $\pm$ 0.008	0.000	-0.356 $\pm$ 0.009	0.000	0.345 $\pm$ 0.005	0.000

Reward scaling Since the reward can be arbitrarily shifted and scaled without changing the resulting optimal policy, we need to set the scale for each model to compare them meaningfully. Therefore, in the simulation, we set the scaling of each model to have the smallest ℓ_1 distance to the ground truth reward under the state distribution of the expert batch data. In real-world data where there is no ground truth, we choose the scale to minimize the difference of max and min value in each feature between models to make them be in a similar range. See Supp. F for details.

5. Results

We evaluate our model on two tasks: a simulated sepsis task and a real-world clinical treatment task.

Baselines We compare with the widely-used Max-Margin Apprenticeship learning (MMA) that follows the linear design of the reward. We also compare with the counterfactual version of MMA, CIRL, in our simulations. In addition, we also compare with the AIRL framework that uses the current state s_t instead of our what-if reasoning of the future outcome s_{t+1}. Within both AIRL and CAIRL frameworks, we compare 3 reward models: (1) Linear, (2) Node-GAM (GAM), and (3) Fully-Connected Neural Network (FCNN).

5.1. Sepsis: a clinically-motivated simulator with ground truth reward and controlled dynamics

We first experiment on a challenging sepsis simulation environment from Oberst & Sontag (2019). This is a coarse physiological model for sepsis with 4 time-varying vitals (Systolic BP, Percentage of Oxygen, ...) that's discretized (e.g. "low"/"normal"/"high"). Combined with 3 different binary treatments (total 8 actions) and 1 static variable (diabetes), our resulting MDP consists of 1440 possible discrete states. Trajectories are at most 20 timesteps. In addition to 4 vitals, in our feature space, we include a uniform noise feature to make it harder. Since this simulator is a discrete environment, we can solve the exact optimal policy via value iteration to generate expert data. We also use the underlying MDP as our transition model that resembles a perfectly trained counterfactual model. We generate 5000 trajectories with optimal policy for both training and test data.

To test if our model can recover the ground truth reward, we design the reward function in two forms. (1) GAM MDP: we model the reward as an additive function of s_{t+1}, i.e. $r = \sum_j f_j(s_{(t+1)j})$. (2) Linear MDP: we model the reward as a linearly additive function of s_{t+1} i.e. $r = \boldsymbol{w} \cdot \boldsymbol{s}_{t+1}$. Its specific functional form can be found in Supp. C. Note that the rewards may not be clinically meaningful, but they allow us to quantitatively compare different methods.

In Table 2, we show the reward and the distance to ground truth reward of all 8 models under $\gamma = 0.9$ and 0.5. First, in GAM MDP where ground truth is non-linear, we see MMA, CIRL and Linear expectedly perform poorly because of their linear nature. Out of all models, GAM-CAIRL achieves the highest reward and also recovers ground truth more faithfully with the lowest distance on the shape graph to the ground truth (Fig. 3). It also outperforms GAM-AIRL, which does not include what-if reasoning but still achieves reasonable performance. In Linear MDP, we see Linear-CAIRL performs the best. Linear-AIRL has a similar reward but its distance on the graph is much larger (0.051 v.s. 0.016). MMA and CIRL perform slightly worse than Linear-AIRL and Linear-CAIRL. GAM and FCNN perform well without significant differences from Linear-CAIRL.

We repeat the experiment with $\gamma = 0.5$ and find the reward difference between models becomes smaller. In GAM MDP, GAM-CAIRL still achieves the smallest distance to the

Table 3. The test accuracy (with stdev) of actions matched to the expert in the MIMIC3. BC is behavior cloning.

	BC	Linear-AIRL	GAM-AIRL	FCNN-AIRL	Linear-CAIRL	GAM-CAIRL	FCNN-CAIRL
Action Acc(%)	72.0 ± 1.0	74.1 ± 0.4	74.2 ± 0.9	73.8 ± 0.5	**74.9** ± 0.4	**74.7** ± 0.3	74.4 ± 0.4

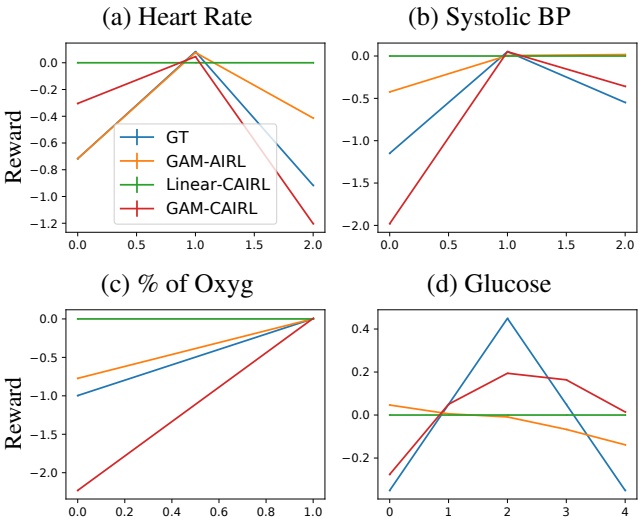

Figure 3. The 4 shape plots in the sepsis dataset where rewards are modeled by a GAM model. Our GAM-CAIRL (red) is closest to the ground truth (GT, blue), while Linear (green) model can not handle non-linear reward and thus act as a straight line.

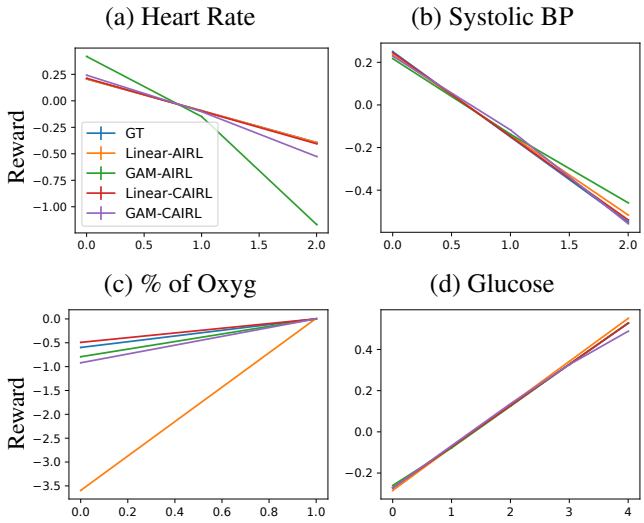

Figure 4. The 4 shape plots in the sepsis dataset where rewards are modeled by a Linear model. All 4 models are close to GT (Blue) except the GAM-AIRL (green) in (a) HR and the Linear-AIRL (orange) in % of Oxyg.

ground truth. In Linear MDP, however, GAM has a smaller distance than Linear in both CAIRL and AIRL settings; we find Linear has an opposite slope in feature Glucose that leads to a larger distance.

We visualize our shape graphs in GAM MDP when $\gamma = 0.9$ in Fig. 3. First, Linear as expected can not capture the non-linear relationship and thus is flat. In (a) Heart Rate, (b) Systolic BP and (c) % of Oxygen, all models except Linear capture the correct trend. For (d) Glucose, only GAM-CAIRL captures the correct shape that finds value 2 produces the highest reward. In Fig. 4, we show the shape graphs in Linear MDP. All models capture the correct trend in all 4 features.

5.2. MIMIC3 Hypotension Treatment Dataset

To demonstrate the utility of our method, we experiment on a real-world medical decision making task of managing hypotensive patients in the ICU. Hypotension is correlated with high mortality (Jones et al., 2004). Although there exists various clinical guidelines (Bunin; Khanna, 2018), there is no standardized treatment strategy since there are many underlying causes of hypotension (Srinivasan & Doshi-Velez, 2020).

Preprocessing We use MIMIC-III (Johnson et al., 2016), filtering to adult patients with at least 2 treatments within the first 72 hours into ICU resulting in 9,404 ICU stays. We discretize trajectories into 2-hour windows, so trajectories end either at ICU discharge or at 72 hours into the ICU admission with at most 36 timesteps and 35 actions taken. We follow the preprocessing of Futoma et al. (2020) to select two treatments: fluid bolus therapy and vasopressors. We discretize both treatments into 4 levels (none, low, medium and high). We extract 5 covariates and 29 time-varying features and impute missing values with the forward imputation. For each model, we perform 5-fold cross validation with each fold having 60-20-20 for train-val-test splits. We set $\gamma = 1$. More details are in Supp. B.

In Table 3, we compare the accuracy of the actions matched to the expert. Note that this only evaluates how good the policy matches experts under experts' states distributions and not the actual, unknown reward, so it's only a proxy of how good the policy is. We also compare with behavior cloning (BC) which does the supervised learning from the logged expert data. Both Linear-CAIRL and GAM-CAIRL perform the best and outperform BC and their AIRL counterparts.

In Fig. 5, we evaluate the shape graphs derived from both our Linear-CAIRL and GAM-CAIRL. First, in Fig. 6, we show the clinician-designed reward for treating hypotensive

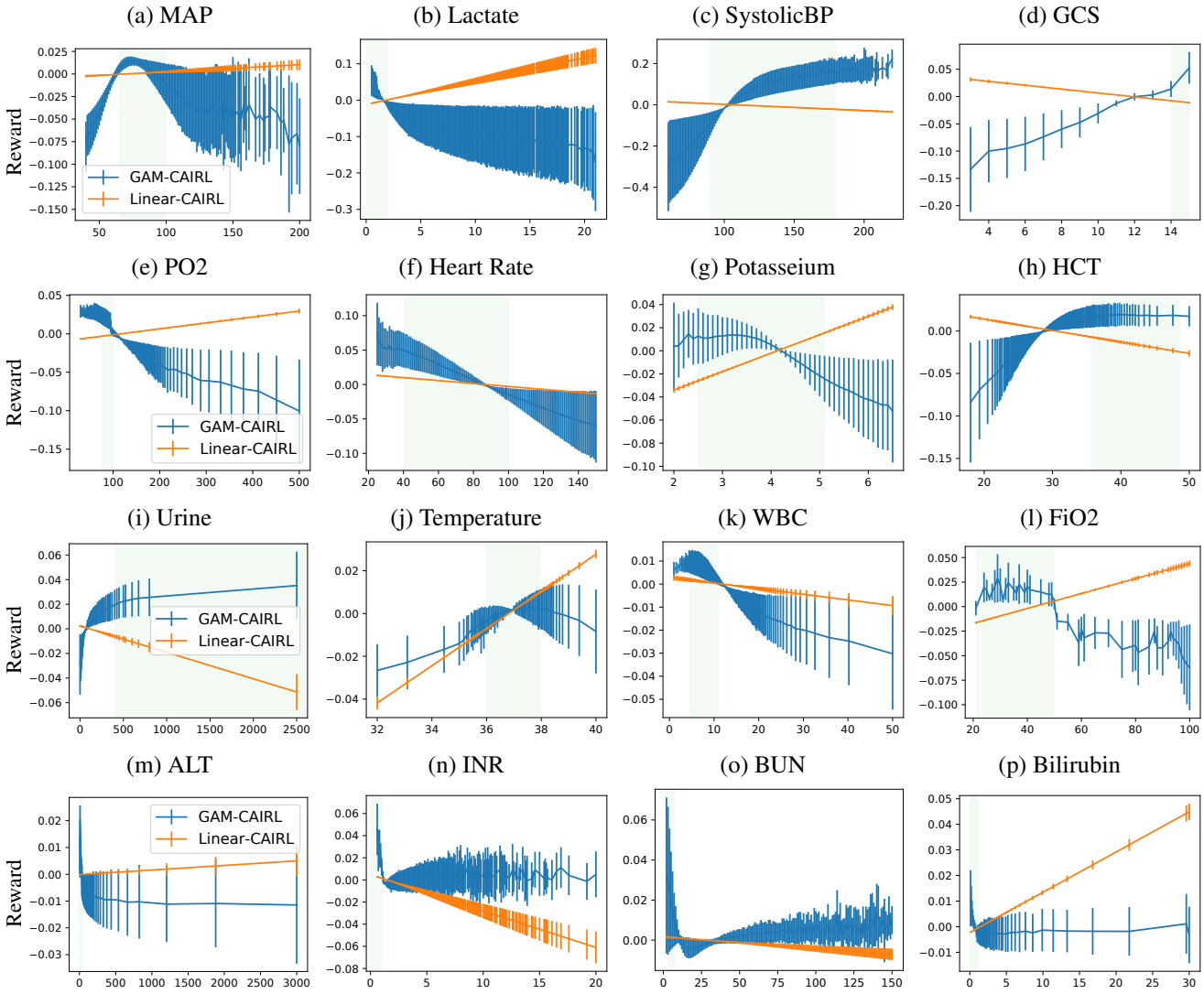

Figure 5. The 16 (out of 29) shape plots of GAM-CAIRL v.s. Linear-CAIRL on MIMIC3 Hypotension management tasks. The pale green region shows the normal range of values based on clinical guidelines (Supp. G).

patients (Futoma et al., 2020) as our ground truth of two features: MAP and Lactate. We find GAM-CAIRL recovers the right regions: in Fig. 5(a) the reward increases as MAP increases above 65 and decreases after 100, which matches the normal range of MAP between 65 and 100 in Fig. 6. Similarly, in Fig. 5(b) the reward of lactate substantially drops as it grows beyond the value 2 and keeps slowly decreasing matching the trend in Fig. 6. Despite the fact that Linear-CAIRL has similar accuracy to GAM-CAIRL (Table 3), it only learns a modest increase in MAP, and an opposite trend in Lactate which is counter to clinical intuitions.

We illustrate other features (c)-(h). In Systolic BP (c), since the goals of both fluids and vasopressors management are to increase blood pressure, it makes sense that the lower blood pressure has a lower reward. Unfortunately, GAM assigns high reward to high Systolic BP even when it is around 200 which we think is an artifact due to the inductive bias of tree-based Node-GAM that remains flat. Linear model instead indicates a negative slope that suggests the lower the blood pressure the better which is clearly in violation of the goal of treating hypotension. Glasgow Coma Scale (GCS, (d)) describes the level of consciousness of patients with value 15 meaning high consciousness and 3 meaning deep coma. Our GAM model captures this notion by learning a steady increase of reward as GCS increases, while the linear model learns the opposite trend which does not make sense. PO2 (e) measures the oxygen concentration in blood. Studies show that PO2 $>$ 100 leads to hyperoxemia which is

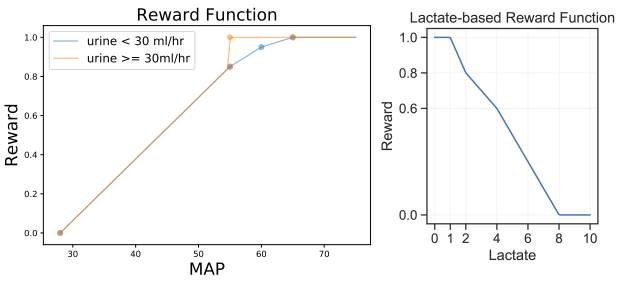

Figure 6. The rewards used in Futoma et al. (2020).

associated with higher mortality. Shape plots show a sharp decrease when PO2 is right above 100 which confirms this. It is important to note that low PO2 (<80) should not be rewarded either, suggesting that the high reward learned by GAM at PO2<80 is likely another artifact. Again, Linear model learns the completely opposite trend in PO2. For heart rate (f), both GAM and Linear model correctly agree that slower heart rate is generally better although the heart rate of 20 is likely too low and should not be rewarded. For potassium (g), high potassium is correlated with kidney disease and sometimes can cause a heart attack or death. Our GAM roughly matches the clinical guideline that assigns a higher reward for normal range (2.5-5.1) and assigns a much lower reward for high potassium. Instead linear model has the opposite trend again. In (h), hematocrit level (HCT) is the percentage of red cells in the blood, and normally when hematocrit is too low it indicates an insufficient supply of healthy red blood cells (anemia). GAM successfully captures this by assigning high reward in HCT but Linear model again contradicts the clinical knowledge.

In the third row (i)-(l), urine volume (i) is correlated with blood pressure and usually high output is a good sign of health while low volume (<50) is an indicator of acute or chronic kidney disease. GAM successfully captures this trend by learning much lower reward for low urine especially below 50, while Linear model learns a higher reward for lower urine output. Body temperature (j) should be maintained between 36-38 degrees. Values that are higher or lower are concerning; GAM captures it perfectly while Linear model learns the upward trend that higher the temperature the better, failing to capture the needed non-linearity. WBC (k) has a normal range between $4.5 - 11$, and high WBC often indicates an infection. GAM displays a steady decrease once the threshold of 10 is exceeded. FiO2 (l) is usually maintained below 50 even when ventilation is used to avoid oxygen toxicity, and we clearly see a sharp decrease of reward above 50 in GAM, but Linear model again learns a counter-intuitive trend.

In the last row (m)-(p), ALT (m) is an enzyme found in the liver, and a high ALT value implies damaged liver that releases ALT into the bloodstream. GAM captures this and prefers the lower value and quickly decreases reward when value exceeds 50, matching the clinical guideline. And again Linear model learns the opposite. INR (n) is a prothrombin time (PT) test that measures the time it takes for the liquid portion of one's blood to clot, and normal people have values below 1.1. GAM captures this threshold by modeling a sharp decrease of the reward after 1.1, while Linear model is unable to learn this threshold effect. BUN (o) measures the amount of urea nitrogen in the blood, and a high BUN level implies worse conditions like heart failure or shock. Both GAM and Linear model capture the correct downturn trend but GAM learns a sharp drop around 8 similar to the clinical guideline. Finally, Bilirubin (p) is a yellow pigment that occurs normally when part of one's red blood cells break down. High bilirubin levels are a sign that the liver isn't clearing the bilirubin from one's blood as it should. GAM again captures the important clinical threshold of 1.2 while the Linear model has the opposite trend learning that higher BUN is better.

6. Discussions and Conclusions

Our framework necessitates a good counterfactual transition model. If the transition model has not been properly tuned, the results might not be meaningful, especially in healthcare measurements are usually long-tailed with missing values. We find quantile transformation is crucial to avoid outliers; training with ℓ_1 loss instead of ℓ_2 also improves the model.

Parameterizing reward functions using counterfactuals gives us an explanation of which outcomes experts likely prefer. As the reward could be sparse or dense, or depends on the current state or next state in different environments, we do not claim our use of counterfactuals always recovers exactly the underlying expert reward. Rather we simply provide an interpretable way to explain how decision-makers are behaving in terms of future counterfactuals.

We observed that linear models often generated the opposite rewards from GAMs. We believe it happens because in the process of trying to fit data that is not linear it compensates by changing the signs of other correlated features thus creating counter-intuitive patterns.

The discretization in RL needs to be done carefully. Vasopressors and fluids should take effect within 2 hours, thus we discretized our data in 2-hour windows. If treatments take longer to have an effect, a different discretization or considering multiple timesteps ahead might be needed.

In this work, we propose to explain clinician's rewards by future counterfactuals and show our GAM explanations match clinical guidelines better than the CIRL in a difficult, batch real-world clinical dataset. Although we do not provide theoretical analysis, we note that CAIRL has the same theoretical convergence as AIRL if we assume the rewards

are of GAM form and conditioned on future states, and all the causal assumptions are valid. We envision our work to help RL practitioners better design human-aligned and safer rewards in a high-stakes setting such as healthcare.

References

Abbeel, P. and Ng, A. Y. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Bica, I., Jarrett, D., Hüyük, A., and van der Schaar, M. Learning" what-if" explanations for sequential decision-making. *arXiv preprint arXiv:2007.13531*, 2020.

Bunin, J. Diagnosis and management of hypotension and shock in the intensive care unit.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *KDD*, 2015.

Chang, C.-H., Mai, M., and Goldenberg, A. Dynamic measurement scheduling for event forecasting using deep rl. In *International Conference on Machine Learning*, pp. 951–960. PMLR, 2019.

Chang, C.-H., Caruana, R., and Goldenberg, A. Node-gam: Neural generalized additive model for interpretable deep learning. *arXiv preprint arXiv:2106.01613*, 2021a.

Chang, C.-H., Tan, S., Lengerich, B., Goldenberg, A., and Caruana, R. How interpretable and trustworthy are gams? In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 95–105, 2021b.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Fatemi, M., Killian, T. W., Subramanian, J., and Ghassemi, M. Medical dead-ends and learning to identify high-risk states and treatments. *Advances in Neural Information Processing Systems*, 34, 2021.

Futoma, J., Hughes, M. C., and Doshi-Velez, F. Popcorn: Partially observed prediction constrained reinforcement learning. *arXiv preprint arXiv:2001.04032*, 2020.

Haarnoja, T., Tang, H., Abbeel, P., and Levine, S. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pp. 1352–1361. PMLR, 2017.

Hastie, T. and Tibshirani, R. Generalized additive models for medical research. *Statistical methods in medical research*, 4(3):187–196, 1995.

Hegselmann, S., Volkert, T., Ohlenburg, H., Gottschalk, A., Dugas, M., and Ertmer, C. An evaluation of the doctor-interpretability of generalized additive models with interactions. In *Machine Learning for Healthcare Conference*, 2020.

Ho, J. and Ermon, S. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

Jenni, S. and Favaro, P. On stabilizing generative adversarial training with noise. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12145–12153, 2019.

Johansson, F., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In *International conference on machine learning*, pp. 3020–3029. PMLR, 2016.

Johnson, A. E., Pollard, T. J., Shen, L., Li-Wei, H. L., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

Jones, A. E., Aborn, L. S., and Kline, J. A. Severity of emergency department hypotension predicts adverse hospital outcome. *Shock*, 22(5):410–414, 2004.

Kaur, H., Nori, H., Jenkins, S., Caruana, R., Wallach, H., and Wortman Vaughan, J. Interpreting interpretability: Understanding data scientists' use of interpretability tools for machine learning. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, pp. 1–14, 2020.

Khanna, A. K. Defending a mean arterial pressure in the intensive care unit: Are we there yet? *Annals of intensive care*, 8(1):1–2, 2018.

Klein, E., Geist, M., and Pietquin, O. Batch, off-policy and model-free apprenticeship learning. In *European Workshop on Reinforcement Learning*, pp. 285–296. Springer, 2011.

Lee, B. K., Lessler, J., and Stuart, E. A. Improving propensity score weighting using machine learning. *Statistics in medicine*, 29(3):337–346, 2010.

Lee, D., Srinivasan, S., and Doshi-Velez, F. Truly batch apprenticeship learning with deep successor features. *arXiv preprint arXiv:1903.10077*, 2019.

Lim, B., Alaa, A. M., and van der Schaar, M. Forecasting treatment responses over time using recurrent marginal structural networks. *NeurIPS*, 18:7483–7493, 2018.

Nori, H., Jenkins, S., Koch, P., and Caruana, R. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

Oberst, M. and Sontag, D. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pp. 4881–4890. PMLR, 2019.

Qureshi, A. H., Boots, B., and Yip, M. C. Adversarial imitation via variational inverse reinforcement learning. *arXiv preprint arXiv:1809.06404*, 2018.

Robins, J. M., Hernan, M. A., and Brumback, B. Marginal structural models and causal inference in epidemiology, 2000.

Rosenbaum, P. R. and Rubin, D. B. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Rubin, D. B. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pp. 34–58, 1978.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016.

Sallab, A. E., Abdou, M., Perot, E., and Yogamani, S. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging*, 2017(19):70–76, 2017.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

Schulam, P. and Saria, S. Reliable decision support using counterfactual models. *Advances in Neural Information Processing Systems*, 30:1697–1708, 2017.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Srinivasan, S. and Doshi-Velez, F. Interpretable batch irl to extract clinician goals in icu hypotension management. *AMIA Summits on Translational Science Proceedings*, 2020:636, 2020.

Stuart, E. A. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1): 1, 2010.

Tan, S., Caruana, R., Hooker, G., and Lou, Y. Distill-and-compare: Auditing black-box models using transparent model distillation. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 303–310, 2018.

Tan, S., Caruana, R., Hooker, G., Koch, P., and Gordo, A. Learning global additive explanations of black-box models. 2019.

Yang, Y., Morillo, I. G., and Hospedales, T. M. Deep neural decision trees. *arXiv preprint arXiv:1806.06988*, 2018.

Ziebart, B. D., Maas, A. L., Bagnell, J. A., Dey, A. K., et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pp. 1433–1438. Chicago, IL, USA, 2008.

A. Causal assumptions

Here we illustrate the causal assumptions we make in order to learn the correct potential outcomes under observstional patient data for the counterfactual transition model. Why do we need these assumptions? This is because we have only access to a batch clinical dataset (e.g. electronic health records) in which the outcomes of patients are entangled with the treatments they receive. For example, seriously ill patients are more likely to get aggressive drug treatments, but they are also more likely to have adverse outcomes. A model without corrections might wrongly conclude that the drug treatments lead to adverse outcomes.

To identify the counterfactual outcomes from observational data we make the standard assumptions of consistency, positivity and no hidden confounders as described in Assumption 1 (Rosenbaum & Rubin, 1983; Robins et al., 2000; Bica et al., 2020). Under Assumption 1, we can estimate potential outcome $E[Y_{t+1}[a_t]|h_t] = E[X_{t+1}|a_t, h_t]$ by training a regression model on the batch observational data.

Assumption 1 (Consistency, Ignorability and Overlap). For any individual i, if action a_t is taken at time t, we observe $X_{t+1} = Y_{t+1}[a_t]$. Moreover, we have sequential strong ignorability (no hidden confounder) assumption that $\{Y_{t+1}[a]_{a\in A}\} \perp a_t|h_t$ for any t, and sequential overlap $\Pr(A_t = a|h_t) > 0$ for all a, t.

Clinically, the no hidden confounders means that we observe all features affecting the action assignment and outcomes. Sequential overlap means that at each timestep, every action has a non-zero probability in the observed batch data which can be satisfied by non-deterministic expert policy. These assumptions are standard across causal inference methods (Schulam & Saria, 2017; Lim et al., 2018). If these assumptions are valid, we can learn an unbiased model by either matching (Stuart, 2010), propensity weighting (IPTW, Lee et al. (2010)), or adversarial representation learning (Johansson et al., 2016). In this paper, we follow the time-series version of propensity weighting (Lim et al., 2018) to learn an unbiased model.

B. MIMIC3 Preprocessing

We follow Futoma et al. (2020) to extract 5 covariates, 29 time-varying features and 10 features related to actions. We use the quantile transformation to Gaussian distribution and finds models provide more meaningful results than the log transformation used in Futoma et al. (2020). For action features, we calculate the past treatment values including the treatment in the last time point (e.g. last_fluid_2 means if the patient gets treated in the last time point with value 2 (medium)), and the treatment value in the last 8 hours, and the total treatment values so far. We also include the missingness indicator for each feature since medical data is not missing at random which results in total 73 features.

- covariates: age, is_F, surg_ICU, is_not_white, is_emergency, is_urgent
- features: dbp, fio2, hr, map, sbp, spontaneousrr, spo2, temp, urine, weight, bun, magnesium, platelets, sodium, alt, hct, po2, ast, potassium, wbc, bicarbonate, creatinine, lactate, pco2, bilirubin_total, glucose, inr, hgb, GCS
- action features: last_vaso_1, last_vaso_2, last_vaso_3, last_fluid_1, last_fluid_2, last_fluid_3, total_all_prev_vasos, total_all_prev_fluids, total_last_8hrs_vasos, total_last_8hrs_fluids

C. Sepsis simulation reward design

In GAM MDP, we simulate the reward using the following state value pair:

- Heart rate: {0: -0.8, 1: 0, 2: -1}
- Systolic BP: {0: -1.2, 1: 0, 2: -0.6}
- % of Oxygen: {0: -1, 1: 0}
- Glucose: {0: -0.8, 1: -0.4, 2: 0, 3: -0.4, 4: -0.8}

In Linear MDP, we simulate the reward using the following state value pair:

- Heart rate: {0: -0.3, 1: -0.6, 2: -0.9}

Table 4. Hyperparameters for GRU training for Behavior Cloning (BC) and Transition Model. We use random search to find the best hyperparameters.

	GRU BC	GRU Transition Model
Epochs	200	200
Batch size	64, 128, 256	128, 256
Learning Rate	5e-4, 1e-3, 2e-3	5e-4, 1e-3
Weight Decay	0, 1e-6, 1e-5, 1e-4	0, 1e-5
GRU Num Hidden	64, 128, 256	64
GRU Num layers	1	1
GRU Dropout	0.3, 0.5	0.3, 0.5
FC Num Hidden	128, 256, 384, 512	256, 384, 512
FC Num Layers	2, 3, 4	2
FC Dropout	0.15, 0.3, 0.5	0.15
FC Activation	ELU	ELU
Act Num Hidden	-	64, 128
Act Num Layers	-	0, 1, 2
Act Num output	-	32, 64, 96
Act Dropout	-	0.3

- Systolic BP: {0: -0.4, 1: -0.8, 2: -1.2}
- % of Oxygen: {0: 0, 1: 0.6}
- Glucose: {0: 0, 1: 0.2, 2: 0.4, 3: 0.6, 4: 0.8}

D. GRU training for behavior cloning and counterfactual transition model

Behavior Cloning (BC) Behavior cloning model takes in the history h_t to predict the current action a_t in the expert batch data. We use GRU to model the prediction. So we feed the history h_t into the GRU to produce output o_t, and then feed the output into several layers of fully-connected layers (FC) with dropout and batchnorm to produce the final classification of action a_t. We list the hyperparameters in Table 4.

Counterfactual Transition Model Learning Counterfactual models take in the current history h_t and action a_t to predict the next states s_{t+1}. We use the similar architecture as the GRU for behavior cloning, except we also use action a_t as inputs and do the regression to predict s_t. Specifically, for action a_t, we go through several layers of Fully Connected Layers to produce action embeddings ϕ_a, and concatenate with the state s_t as the inputs to the GRU model. For output, we use the Huber loss (smoothed ℓ_1 loss) that we find it produces more diverse states than Mean Squared Error (MSE) loss. Finally, we weight the samples by stabilized Inverse Propensity Weighting (IPTW) that gives different sample weights w_t. We list the hyperparameters in Table 4. Specifically, given the behavior cloning policy π_{bc}, action embedding layers (Act) A, GRU model G and output fully connected layer F, we have

$$
\begin{aligned}
\phi_a &= A(a_t) \\
\phi &= \text{concat}(s_t, \phi_a) \\
g_t &= M(\phi) \\
o_t &= F(g_t) \\
w_t &= \frac{P(a_t)}{\pi_{bc}(a_t)} \\
L &= w_t \cdot \text{Huber}(o_t, s_{t+1}) \\
\theta &\longleftarrow \theta - \nabla_\theta L \quad \text{(Updated by Adam)}
\end{aligned}
$$

Table 5. Hyperparameters for Node-GAM training for discriminator training. We use random search to find the best hyperparameters.

	Simulation	MIMIC3
Epochs	100	100
Input Noise	0	0,0.1
Noise decay	0	80%
LR	2e-4, 4e-4	5e-4, 8e-4, 1e-3
Label Smoothing δ	0	0,0.005,0.01
Num Layers	1, 2	1,2,3
Num Trees	100, 200, 400	200,300,400
Addi Tree Dim	0, 1	0, 1
Depth	1, 2	2,3,4
Output Dropout	0, 0.1	0.1,0.2
Last Dropout	0, 0.3	0.3,0.5
Column Subsample	1, 0.5	0.5
Temp Annealing	3000	3000

E. AIRL training

E.1. Discriminator Training

Here we train a discriminator that can distinguish expert batch data as class 1 and generated experience as class 0 in the binary classification setting. And its logit would represent the expert reward r. Given a batch of expert data X_E, we generate the data X_G by executing the generator policy π in the trained counterfactual transition model T. Note that to explain the expert in terms of future counterfactuals, we exclude the static covariates and action features and only use the time-varying features of next state when training the discriminator.

Discriminator Stablizing Tricks When optimizing discriminator, we use both one-sided label smoothing (Salimans et al., 2016) which reduces the label confidence for the expert batch data. We also add a small input Gaussian noise to the inputs for both expert and generated data, and linearly decayed the noise throughout the training. It has been shown useful to stabilize GAN adversarial optimization (Jenni & Favaro, 2019). Specifically, given label smoothing δ, input noise δ_n, the discriminator model $\mathcal{D}$, expert batch data D_E, the transition model T, the generator policy π_G and binary cross entropy loss (BCE):

$$
\begin{aligned}
&X_E = (s, a, s') \sim D_E \\
&X_G = (s, a, s') \text{ where } s \sim X_E, a \sim \pi_G(\cdot|s) \text{ and } s' \sim T(\cdot|s, a) \\
&n \sim \mathcal{N}(\mathbf{0}, \mathbf{1}) \\
&X_E = X_E + \delta_n \cdot n,\ X_G = X_G + \delta_n \cdot n \\
&L = \text{BCELoss}(D(X_E), \mathbf{1} - \delta) + \text{BCELoss}(D(X_G), \mathbf{0})
\end{aligned}
$$

We list the hyperparameters we use for Node-GAM in Table 5. And we list the linear and FCNN model's hyperparameters in Table 6.

E.2. Generator Training

In Sepsis simulation, since we use the value iteration to solve the exact optimal policy for our generator, there is no hyperparameter to tune. To save the computation, we only update the generator after the discriminator updates for 20 steps.

In MIMIC3, we use the soft-Q learning as our generator as described in Sec. 4. We use a fully connected neural net with dropout, ELU activation function and batchnorm as our architecture. We show the hyperparameters in Table 7.

F. Reward scaling

Since the reward can be arbitrarily shifted and scaled without changing the resulting optimal policy, comparing the reward across models requires setting the scale of the reward for each model when showing the GAM plots and calculating distance.

Table 6. Hyperparameters for Linear model and FCNN model for discriminator training.

	Linear		FCNN	
	Simulation	MIMIC3	Simulation	MIMIC3
Epochs	100	100	100	100
Input Noise	0	0,0.1	0	0,0.1
Noise decay	0	0.8	0	0.8
LR	2e-4, 4e-4	5e-4, 8e-4, 1e-3	2e-4, 4e-4	5e-4, 8e-4, 1e-3
Label Smoothing δ	0	0,0.005,0.01	0	0,0.005,0.01
Num Layers	-	-	2,3,4,5	2,3,4,5
Num Hidden	-	-	32,64,128,256	32,64,128,256
Dropout	-	-	0.1,0.3,0.5	0.1,0.3,0.5

Table 7. Hyperparameters for generator training in Sepsis and MIMIC3 datasets.

	Sepsis	MIMIC3
Update Freq	20	1
Epochs	-	100
Entropy Coeff α	-	0.25, 0.5
Sample weights for gen data (δ)	-	0.5
Sync Rate	-	200
LR	-	4e-4, 8e-4
Num Layer	-	3, 4
Dropout	-	0.3,0.5
BC Reg	-	10
BC Reg decay	-	0.5

Therefore, in the simulation for each model, we shift the average reward to 0 and set the scaling a that has the smallest ℓ_1 distance to the ground truth reward under the state distribution of the expert batch data. Given the ground truth model G and its GAM main effect $f_G(x_j)$ of each feature j, model M and $f_M(x_j)$, V_j as all the values of feature j, with each value $v \in V_j$, and the counts $c(v)$ in the expert batch data, we derive a by convex optimization:

$$\min_a \sum_{j=1}^{D} \sum_{v \in V_j} |(f_G(v) - a f_M(v))| c(v)$$

In real-world data where there is no ground truth, we choose the scale a to minimize the difference of max and min value in each feature of two models G, M to make them display in the similar range:

$$\min_a \sum_{j=1}^{D} (|\min_{v \in V_{Gj}} f_G(v) - a \min_{v \in V_{Mj}} f_M(v)| + |\max_{v \in V_{Gj}} f_G(v) - a \max_{v \in V_{Mj}} f_M(v)|).$$

G. Clinicial guidelines sources

Here we list the lower and upper bound, and the sources of the normal ranges we use in Fig. 5.

- MAP: (70, 100) https://www.healthline.com/health/mean-arterial-pressure#:~:text=What%20is%20a%20normal%20MAP,100%20mmHg%20to%20be%20normal.
- Lactate: (0, 2)
- Systolic BP: upper bound 180 https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3704960/ and lower bound 90 https://www.mayoclinic.org/diseases-conditions/low-blood-pressure/

symptoms-causes/syc-20355465#:~:text=What's%20considered%20low%20blood%20pressure,pressure%20is%20lower%20than%20normal..

- Bicarbonate: (23, 30) https://www.urmc.rochester.edu/encyclopedia/content.aspx?contenttypeid=167&contentid=bicarbonate#:~:text=Normal%20bicarbonate%20levels%20are%3A,30%20mEq%2FL%20in%20adults
- pO2: (75, 100) https://www.medicalnewstoday.com/articles/322343#:~:text=Most%20healthy%20adults%20have%20a,emphysema
- Heart Rate: (40, 100) https://health.clevelandclinic.org/is-a-slow-heart-rate-good-or-bad-for-you
- Potassium: (2.5, 5.1) https://my.clevelandclinic.org/health/diseases/17740-low-potassium-levels-in-your-blood-hypokalemia
- HCT: (35.5, 48.6) https://www.mayoclinic.org/tests-procedures/hematocrit/about/pac-20384728
- Urine: (400, inf) https://www.healthline.com/health/urine-output-decreased#:~:text=Oliguria%20is%20considered%20to%20be,is%20considered%20to%20be%20anuria.
- WBC: (4.5, 11) https://my.clevelandclinic.org/health/diagnostics/17704-high-white-blood-cell-count
- FiO2: (21, 50) https://en.wikipedia.org/wiki/Fraction_of_inspired_oxygen#:~:text=Natural%20air%20includes%2021%25%20oxygen,to%20100%25%20is%20routinely%20used.
- ALT: (0, 55) https://www.mayoclinic.org/tests-procedures/liver-function-tests/about/pac-20394595
- INR: (0, 1.1) https://my.clevelandclinic.org/health/diagnostics/17691-prothrombin-time-pt-test
- BUN: (2.1, 8.5) https://www.mayoclinic.org/tests-procedures/blood-urea-nitrogen/about/pac-20384821
- Bilirubin total: (0, 1.2) https://www.webmd.com/a-to-z-guides/bilirubin-test

H. Computing Resources Used

All experiments are run on 1 P100 GPU, 4 CPU and 16G RAM on a cluster.

I. Complete shape graphs

We show the complete shape graphs of 29 features in MIMIC-III in Fig. 7.

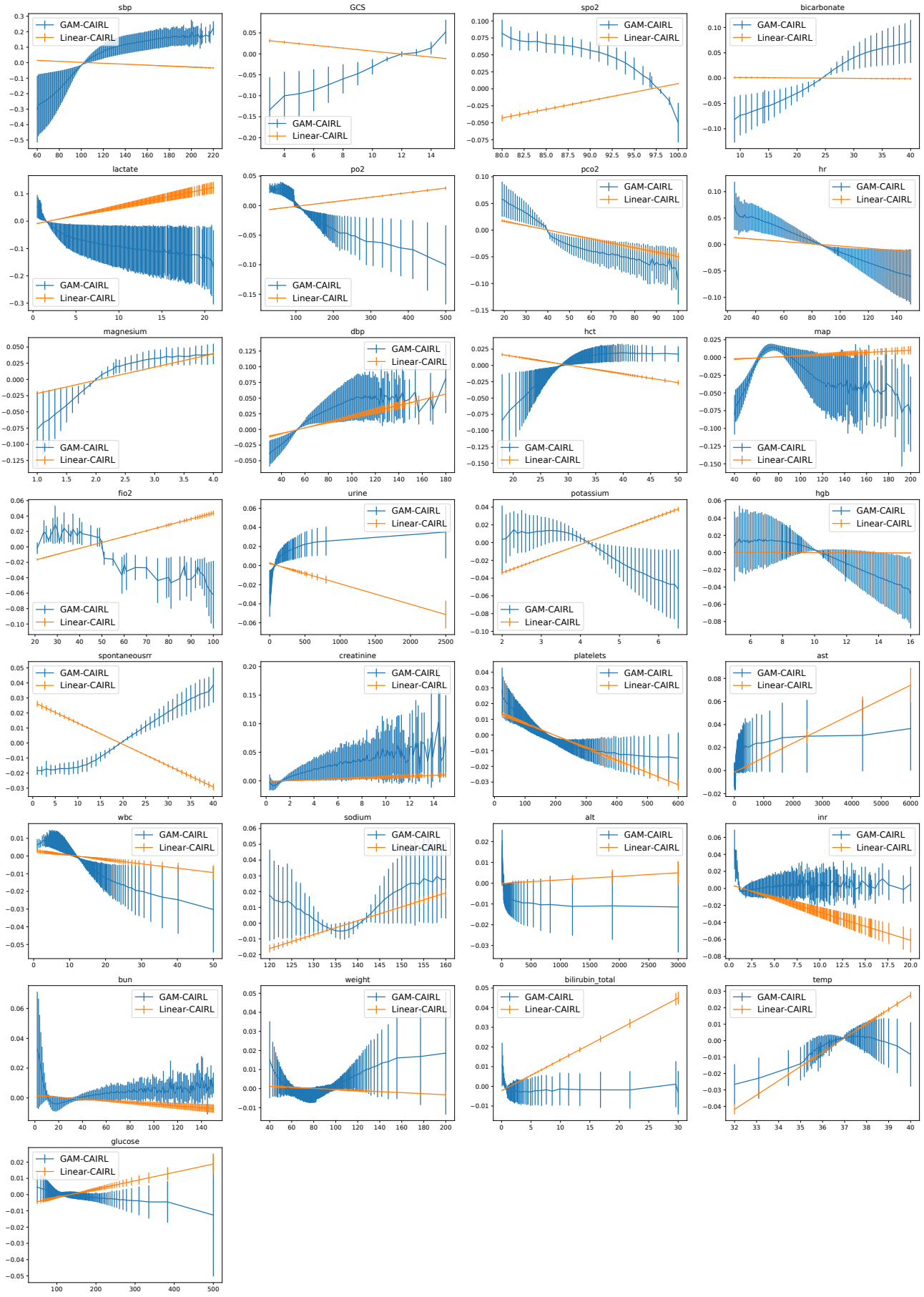

Figure 7. The complete shape plots of MIMIC3.