Estimating the Effects of Continuous-valued Interventions using Generative Adversarial Networks

Ioana Bica* IOANA.BICA@ENG.OX.AC.UK
Department of Engineering Science
University of Oxford, Oxford, United Kingdom
The Alan Turing Institute, London, United Kingdom

James Jordon* JAMES.JORDON@WOLFSON.OX.AC.UK
Department of Engineering Science
University of Oxford, Oxford, United Kingdom

Mihaela van der Schaar MV472@CAM.AC.UK
University of Cambridge, Cambridge, UK
University of California, Los Angeles, USA
The Alan Turing Institute, London, UK

Abstract

While much attention has been given to the problem of estimating the effect of discrete interventions from observational data, relatively little work has been done in the setting of continuous-valued interventions, such as treatments associated with a dosage parameter. In this paper, we tackle this problem by building on a modification of the generative adversarial networks (GANs) framework. Our model, SCIGAN, is flexible and capable of simultaneously estimating counterfactual outcomes for several different continuous interventions. The key idea is to use a significantly modified GAN model to learn to generate counterfactual outcomes, which can then be used to learn an inference model, using standard supervised methods, capable of estimating these counterfactuals for a new sample. To address the challenges presented by shifting to continuous interventions, we propose a novel architecture for our discriminator - we build a hierarchical discriminator that leverages the structure of the continuous intervention setting. Moreover, we provide theoretical results to support our use of the GAN framework and of the hierarchical discriminator. In the experiments section, we introduce a new semi-synthetic data simulation for use in the continuous intervention setting and demonstrate improvements over the existing benchmark models.

Keywords: continuous interventions, causal inference, treatment effects, generative adversarial networks

1. Introduction

Estimating the personalised effects of interventions is crucial for decision making in many domains such as medicine, education, public policy and advertising. Such domains have a wealth of observational data available. Most of the methods developed in the causal inference literature focus on learning the counterfactual outcomes of discrete interventions, such as binary or categorical treatments[1] (Bertsimas et al., 2017; Alaa et al., 2017; Alaa and van der Schaar, 2017; Athey and Imbens, 2016; Wager and Athey, 2018; Yoon et al., 2018). Unfortunately, in many cases, deciding how to intervene involves not only deciding which intervention to make (e.g. whether to treat cancer with radiotherapy, chemotherapy or surgery) but also deciding on the value of some continuous parameter associated with intervening (e.g. the dosage of

*Equal contribution.

[1] For ease of exposition, we will sometimes refer to interventions as treatments and to the associated continuous parameter as the dosage throughout the paper.

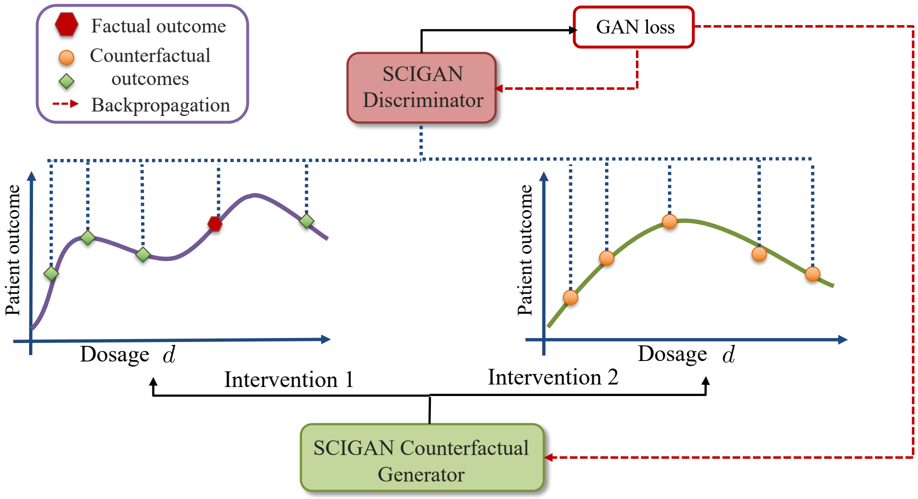


Figure 1: Overview of GAN framework used in SCIGAN for learning the distribution of the counterfactual outcomes.

radiotherapy to be administered). In medicine there are many examples of treatments that are associated with a continuous dosage parameter (such as vasopressors (Döpp-Zemel and Groeneveld, 2013)). In the medical setting, using a high dosage for a treatment can lead to toxic effects while using a low dosage can result in no effect on the patient outcome (Wang et al., 2017). In other domains, there are many examples of continuous interventions, such as the duration of an education or job training program, the frequency of an exposure or the price used in an advert. Naturally, being able to estimate the effect of these continuous interventions will aid in the decision making process.

Learning from observational data already presents significant challenges when there is only a single intervention (and thus the decision is binary - whether to intervene or not). As explained in Spirtes (2009), in an observational dataset, only the factual outcome is present - the “counterfactual" outcomes are not observed. This problem is exacerbated in the setting of continuous interventions where the number of counterfactuals is no longer even finite. Moreover, the decision to intervene is non-random and instead is assigned according to the features associated with each sample. Due to the continuous nature of the interventions, adjusting for selection bias is significantly more complex than for binary (or even multiple) interventions. Thus, standard methods for adjusting for selection bias for discrete treatments cannot be easily extended to handle bias in the continuous setting.

In this paper we propose SCIGAN (eStimating the effects of Continuous Interventions using GANs). We build on the GAN framework of Goodfellow et al. (2014) in order to learn the distribution of the unobserved counterfactuals. GANs have already been used in GANITE (Yoon et al., 2018) to generate the unobserved counterfactual outcomes for discrete interventions. The intuition is that if a counterfactual generator and discriminator are trained adversarially, then the generator can fool the discriminator (i.e. the discriminator will not be able to correctly identify the factual outcome) by generating counterfactuals according to their true distribution. Unfortunately, no theoretical work was provided in Yoon et al. (2018) to back up this intuition. A key contribution of this paper is to provide theoretical results that justify using the GAN framework to learn to generate counterfactual outcomes; these results also apply to GANITE.

GANITE itself presents a significant modification to the original GAN framework - rather than the discriminator discriminating between entirely real or entirely fake samples, the discriminator is attempting to identify the real component from a vector containing the real (factual) outcome from the dataset and the fake (counterfactual) outcomes generated by the generator. SCIGAN inherits this key difference from a standard GAN. However, beyond our theoretical contribution, we propose significant changes to the generator and discriminator in order to tackle the more complex problem of estimating outcomes of continuous interventions.

Naive attempts to extend Yoon et al. (2018) to the continuous setting might involve: (1) discretising the continuous space of interventions; (2) somehow passing entire response curves to the discriminator and asking it to identify the point on the curve that corresponds to the factual outcome.

Naturally, discretisation comes with a cost. If the discretisation is too coarse, the response curves will not be well-approximated. On the other hand, we show experimentally that GANITE is incapable of handling a high number of discrete interventions (corresponding to having a finer discretisation). In fact, although SCIGAN was designed for continuous interventions, it can be applied in the discrete setting and we show that it outperforms GANITE when the (discrete) parameter space is not small.

For (2), the problem is in defining a mechanism for generating these response curves in a form that can be passed to the discriminator and ensuring the *continuity* of these curves around the factual outcome so that the discontinuity itself does not make identification trivial for the discriminator. To overcome this we define a discriminator that acts on a finite set of points from each generated response curve estimated (rather than on entire curves), as shown in Fig. 1. From *among the chosen points*, the discriminator attempts to identify the factual one. The set of points is sampled randomly *each* time an input would be passed to the discriminator. As our discriminator will be acting on a *set* of random intervention-outcome pairs, we explicitly condition it to behave as a function on a set. In particular, we draw on ideas from Zaheer et al. (2017) to ensure that its output does not depend on the *order* of its input.

In addition, for the setting in which there are multiple possible interventions that *each* have an associated continuous parameter (which is the main setting of the paper), we propose a *hierarchical* discriminator which breaks down the job of the discriminator into determining the factual intervention and determining the factual parameter using separate networks. We show in the experiments section that this approach significantly improves performance and is more stable than using a single network discriminator. In this setting, we also model the generator as a multi-task deep network capable of taking a continuous parameter as an input; this gives us the flexibility to learn heterogeneous response curves for the different interventions.

Our contributions in this paper are 4-fold: (1) we propose SCIGAN, a significantly modified GAN framework, capable of estimating outcomes for continuous and many-level-discrete interventions, (2) we provide theoretical justification for both the use of a GAN framework and a hierarchical discriminator, (3) we propose novel architectures for each of our networks, (4) we propose a new semi-synthetic data simulation for use in the continuous intervention setting. We show, using semi-synthetic experiments, that our model outperforms existing benchmarks.

2. Problem formulation

We consider receiving observations of the form $(\mathbf{x}^i, t_f^i, y_f^i)$ for $i = 1, ..., N$, where, for each i, these are independent realizations of the random variables $(\mathbf{X}, T_f, Y_f)$. We refer to $\mathbf{X}$ as the feature vector lying in some feature space $\mathcal{X}$, containing pre-treatment covariates (such as age, weight and lab test results). The treatment random variable, T_f, is in fact a pair of values $T_f = (W_f, D_f)$ where $W_f \in \mathcal{W}$ corresponds to the *type* of treatment being administered (e.g. chemotherapy or radiotherapy) which lies in the discrete space of k treatments, $\mathcal{W} = \{w_1, ..., w_k\}$, and D_f corresponds to the *dosage* of the treatment (e.g. number of cycles, amount of chemotherapy, intensity of radiotherapy), which, for a given treatment w lies in the corresponding treatment's dosage space, $\mathcal{D}_w$ (e.g. the interval $[0, 1]$). We define the set of all treatment-dosage pairs to be $\mathcal{T} = \{(w, d) : w \in \mathcal{W}, d \in \mathcal{D}_w\}$.

Following Rubin's potential outcome framework (Rubin, 1984), we assume that for all treatment-dosage pairs, (w, d), there is a potential outcome $Y(w, d) \in \mathcal{Y}$ (e.g. 1-year survival probability). The *observed* outcome is then defined to be $Y_f = Y(W_f, D_f)$. We will refer to the unobserved (potential) outcomes as *counterfactuals*.

The goal is to derive *unbiased* estimates of the potential outcomes for a given set of input covariates:

$$\mu(t, \mathbf{x}) = \mathbb{E}[Y(t)|\mathbf{X} = \mathbf{x}] \tag{1}$$

for each $t \in \mathcal{T}$, $\mathbf{x} \in \mathcal{X}$. We refer to $\mu(\cdot)$ as the individualised dose-response function. A table summarising our notation is given in Appendix A. In order to ensure that this quantity is equal to $\mathbb{E}[Y|\mathbf{X} = \mathbf{x}, T = t]$ and that the dose-response function is identifiable from the observational data, we require the following two assumptions.

Assumption 1 *(Unconfoundedness) The treatment assignment, T_f, and potential outcomes, $Y(w, d)$, are conditionally independent given the covariates $\mathbf{X}$, i.e.*

$$\{Y(w, d)|w \in \mathcal{W}, d \in \mathcal{D}_w\} \perp\!\!\!\perp T_f|\mathbf{X}\,. \tag{2}$$

Assumption 2 *(Overlap) For each $\mathbf{x} \in \mathcal{X}$ such that $p(\mathbf{x}) > 0$, we have $1 > p(t|\mathbf{x}) > 0$ for each $t \in \mathcal{T}$.*

3. SCIGAN

We propose estimating μ by first training a generator to generate response curves for each sample *within* the training dataset. The learned generator can then be used to train an inference network using standard supervised methods. We build on the idea presented in Yoon et al. (2018), using a modified GAN framework to generate potential outcomes conditional on the observed features, treatment and factual outcome. Several changes must be made to both the generator and discriminator architectures and learning paradigms in order to produce a model capable of handling the dose-response setting.

3.1 Counterfactual Generator

Our generator, $\mathbf{G} : \mathcal{X} \times \mathcal{T} \times \mathcal{Y} \times \mathcal{Z} \to \mathcal{Y}^{\mathcal{T}}$ takes features, $\mathbf{x} \in \mathcal{X}$, factual outcome, $y_f \in \mathcal{Y}$, received treatment and dosage, $t_f = (w_f, d_f) \in \mathcal{T}$, and some noise, $\mathbf{z} \in \mathcal{Z}$ (typically multivariate uniform or Gaussian), as inputs. The output will be a dose-response curve for each treatment (as shown in Fig. 1), so that the output is a function from $\mathcal{T}$ to $\mathcal{Y}$, i.e. $\mathbf{G}(\mathbf{x}, t_f, y_f, \mathbf{z})(\cdot) : \mathcal{T} \to \mathcal{Y}$. We can then write

$$\hat{y}_{cf}(t) = \mathbf{G}(\mathbf{x}, t_f, y_f, \mathbf{z})(t) \tag{3}$$

as our generated counterfactual outcome for treatment-dosage pair t. We write $\hat{Y}_{cf}(t) = \mathbf{G}(\mathbf{X}, T_f, Y_f, \mathbf{Z})(t)$ (i.e. the random variable induced by $\mathbf{G}$).

While the job of the counterfactual generator is to generate outcomes for the treatment-dosage pairs which were *not* observed, Yoon et al. (2018) demonstrated that the performance of the counterfactual generator is improved by adding a supervised loss term that regularises its output for the factual treatment (in our case treatment-dosage pair). We define the supervised loss, $\mathcal{L}_S$, to be

$$\mathcal{L}_S(\mathbf{G}) = \mathbb{E}\left[(Y_f - \mathbf{G}(\mathbf{X}, T_f, Y_f, \mathbf{Z})(T_f))^2\right], \tag{4}$$

where the expectation is taken over $\mathbf{X}, T_f, Y_f$ and $\mathbf{Z}$.

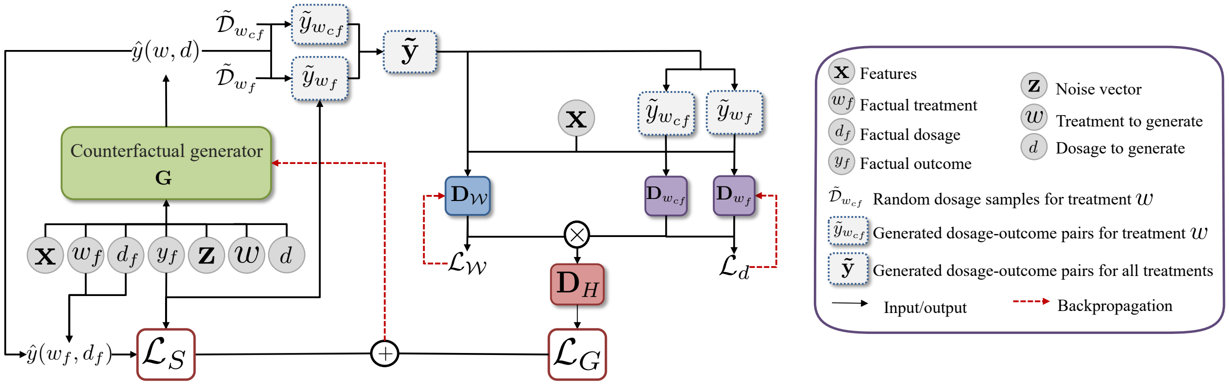


Figure 2: Overview of our model for the setting with two treatments (w_f/w_{cf} being the factual/counterfactual treatment). The generator is used to generate an output for each dosage level in each $\tilde{\mathcal{D}}_w$, these outcomes together with the factual outcome, y_f, are used to create the set of dosage-outcome pairs, $\tilde{\mathbf{y}}$, which is passed to the treatment discriminator. Each dosage discriminator receives only the part of $\tilde{\mathbf{y}}$ corresponding to that treatment, i.e. $\tilde{\mathbf{y}}_w$. These discriminators are combined (Eq. 11) to define $\mathbf{D}_H$ which gives feedback to the generator.

3.2 Counterfactual Discriminator

As noted in Section 1, our discriminator will act on a random set of points from each of the generated dose-response curves. Similar to Yoon et al. (2018), we define a discriminator, $\mathbf{D}$, that will attempt to pick out the factual treatment-dosage pair from among the (random set of) generated ones.

Formally, let $n_w \in \mathbb{Z}^+$ be the number of dosage levels we will compare for treatment $w \in \mathcal{W}$[2]. For each $w \in \mathcal{W}$, let $\tilde{\mathcal{D}}_w = \{D_1^w, ..., D_{n_w}^w\}$ be a random subset[3] of $\mathcal{D}_w$ of size n_w, where for the factual treatment, W_f, $\tilde{\mathcal{D}}_{W_f}$ contains $n_{W_f} - 1$ random elements along with D_f. We define $\tilde{\mathbf{Y}}_w = (D_i^w, \tilde{Y}_i^w)_{i=1}^{n_w} \in (\mathcal{D}_w \times \mathcal{Y})^{n_w}$ to be the vector of dosage-outcome pairs for treatment w where

$$\tilde{Y}_i^w = \begin{cases} Y_f \text{ if } W_f = w \text{ and } D_f = D_i^w \\ \hat{Y}_{cf}(w, D_i^w) \text{ else} \end{cases} \tag{5}$$

and will write $\tilde{\mathbf{Y}} = (\tilde{\mathbf{Y}}_w)_{w \in \mathcal{W}}$. We will write $d_j^w, \tilde{\mathbf{y}}_w$ and $\tilde{\mathbf{y}}$ to denote realisations of $D_j^w, \tilde{\mathbf{Y}}_w$ and $\tilde{\mathbf{Y}}$.

Our discriminator, $\mathbf{D} : \mathcal{X} \times \prod_{w \in \mathcal{W}} (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to [0,1]^{\sum n_w}$, will take the features $\mathbf{x} \in \mathcal{X}$ together with the (random) set of generated outcomes $\tilde{\mathbf{y}} \in \mathcal{Y}^{\sum n_w}$, and output a probability for each treatment-dosage pair indicating the discriminator's belief that that pair is the factual one.

As in the standard GAN framework, we define a minimax game by defining the value function to be

$$\mathcal{L}(\mathbf{D}, \mathbf{G}) = \mathbb{E}\left[\sum_{w \in \mathcal{W}} \sum_{d \in \tilde{\mathcal{D}}_w} \mathbb{I}_{\{T_f = (w,d)\}} \log \mathbf{D}^{w,d}(\mathbf{X}, \tilde{\mathbf{Y}}) + \mathbb{I}_{\{T_f \neq (w,d)\}} \log(1 - \mathbf{D}^{w,d}(\mathbf{X}, \tilde{\mathbf{Y}})) \right], \tag{6}$$

where the expectation is taken over $\mathbf{X}, T_f, \tilde{\mathbf{Y}}$ and $\{\tilde{\mathcal{D}}_w : w \in \mathcal{W}\}$, $\mathbf{D}^{w,d}$ corresponds to the discriminator output for treatment-dosage pair (w, d).

[2] In practice we set all n_w to be the same. The default setting is 5 in the experiments.

[3] In practice, when $\mathcal{D}_w = [0,1]$, each D_j^w is sampled independently and uniformly from $[0,1]$. Note that for each training iteration, $\tilde{\mathcal{D}}_w$ is resampled (see Section 1).

The minimax game is then given by

$$\min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \mathbf{G}) + \lambda \mathcal{L}_S(\mathbf{G})\,, \tag{7}$$

where λ is used to control the trade-off between $\mathcal{L}$ and $\mathcal{L}_S$ (we set λ = 1 in the experiments).

The task of the discriminator (i.e. picking out the factual dosage from $\sum_{j=1}^{k} n_{w_j}$ treatment-dosage pairs) becomes increasingly difficult as we increase n_w or k because the dimension of the discriminator output space, $\sum n_w$, increases. Although we control n_w, if we set it too low, then the set $\tilde{\mathbf{y}}_w$ may not well-represent the dose-response curve, particularly if the dose-response curve is complex. In practice we found that even for moderate settings of n_w and only 2 treatments, modelling the discriminator as a single function resulted in poor performance. In order to overcome this problem, we introduce a novel hierarchical discriminator which involves a treatment discriminator with output dimension k and several dosage discriminators, one for each treatment, with output dimensions n_w.

First observe that the probability $\mathbb{P}((W_f, D_f) = (w, d) | \mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}})$ can be written as

$$\mathbb{P}(W_f = w | \mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}}) \times \mathbb{P}(D_f = d | W_f = w, \mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}})\,. \tag{8}$$

We can therefore break down the discriminator into a hierarchical model by learning one discriminator, $\mathbf{D}_{\mathcal{W}}$, that outputs $\mathbb{P}(W_f = w | \mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}})$ which we will refer to as the *treatment* discriminator, and then a discriminator, $\mathbf{D}_w$, for each treatment, $w \in \mathcal{W}$, that outputs $\mathbb{P}(D_f = d | W_f = w, \mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}})$ which we will refer to as the *dosage* discriminator for treatment w. Note that although these treatment discriminators are different functions, we can model (some of) them as a single network if we believe that the response curves for different treatments are similar or if there is not sufficient data to learn k distinct networks. To do so, we might use a single network that takes an additional input indicating which treatment the discriminator is for. This is a *modelling* choice, and should be driven by knowledge of the problem domain.

The treatment discriminator, $\mathbf{D}_{\mathcal{W}} : \mathbf{X} \times \prod_{w \in \mathcal{W}} (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to [0, 1]^k$, takes the features, $\mathbf{x}$, and generated potential outcomes, $\tilde{\mathbf{y}}$, and outputs a probability for each treatment, $w_1, ..., w_k$. Writing $\mathbf{D}_{\mathcal{W}}^w$ to denote the output of $\mathbf{D}_{\mathcal{W}}$ corresponding to treatment w, we define the loss, $\mathcal{L}_{\mathcal{W}}$, to be

$$\mathcal{L}_{\mathcal{W}}(\mathbf{D}_{\mathcal{W}}; \mathbf{G}) = -\mathbb{E}\left[\sum_{w \in \mathcal{W}} \mathbb{I}_{\{W_f = w\}} \log \mathbf{D}_{\mathcal{W}}^w(\mathbf{X}, \tilde{\mathbf{Y}}) + \mathbb{I}_{\{W_f \neq w\}} \log(1 - \mathbf{D}_{\mathcal{W}}^w(\mathbf{X}, \tilde{\mathbf{Y}}))\right], \tag{9}$$

where, again, the expectation is taken over $\mathbf{X}, W_f, D_f, \tilde{\mathbf{Y}}$ and $\{\tilde{\mathcal{D}}_w\}_{w \in \mathcal{W}}$.

Then, for each $w \in \mathcal{W}$, $\mathbf{D}_w : \mathcal{X} \times (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to [0, 1]^{n_w}$ is a map that takes the features, $\mathbf{x}$, and generated potential outcomes, $\tilde{\mathbf{y}}_w$, corresponding to treatment w and outputs a probability for each dosage level, $d_1^w, ..., d_{n_w}^w$, in a given realisation of $\tilde{\mathcal{D}}_w$. Writing $\mathbf{D}_w^j$ to denote the output of $\mathbf{D}_w$ corresponding to dosage level D_j^w, we define the loss of each dosage discriminator to be

$$\mathcal{L}_d(\mathbf{D}_w; \mathbf{G}) = -\mathbb{E}\left[\mathbb{I}_{\{W_f = w\}} \sum_{j=1}^{n_w} \mathbb{I}_{\{D_f = D_j^w\}} \log \mathbf{D}_w^j(\mathbf{X}, \tilde{\mathbf{Y}}_w) + \mathbb{I}_{\{D_f \neq D_j^w\}} \log(1 - \mathbf{D}_w^j(\mathbf{X}, \tilde{\mathbf{Y}}_w))\right], \tag{10}$$

where the expectation is taken over $\mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}}_w, W_f$ and D_f. The $\mathbb{I}_{\{W_f = w\}}$ term ensures that only samples for which the factual treatment is w are used to train dosage discriminator $\mathbf{D}_w$ (otherwise there would be no factual dosage for that sample).

We define the overall discriminator $\mathbf{D}_H : \mathcal{X} \times \prod_{w \in \mathcal{W}} (\mathcal{D}_w \times Y)^{n_w} \to [0, 1]^{\sum n_w}$ by defining its output corresponding to the treatment-dosage pair (w, d_j^w) as

$$\mathbf{D}_H^{w,j}(\mathbf{x}, \tilde{\mathbf{y}}) = \mathbf{D}_{\mathcal{W}}^w(\mathbf{x}, \tilde{\mathbf{y}}) \times \mathbf{D}_w^j(\mathbf{x}, \tilde{\mathbf{y}}_w)\,. \tag{11}$$

Instead of the minimax game in Eq. 7, the generator and discriminator are trained according to the minimax game defined by seeking $\mathbf{G}^*$, $\mathbf{D}_H^*$ that solve:

$$\begin{aligned} &\mathbf{G}^* = \arg\min_{\mathbf{G}} \mathcal{L}(\mathbf{D}_H^*; \mathbf{G}) + \lambda \mathcal{L}_S(\mathbf{G}) && {\mathbf{D}_H^*}^{w,j} = {\mathbf{D}_{\mathcal{W}}^*}^{w} \times {\mathbf{D}_w^*}^{j} \\ &\mathbf{D}_{\mathcal{W}}^* = \arg\min_{\mathbf{D}_{\mathcal{W}}} \mathcal{L}_{\mathcal{W}}(\mathbf{D}_{\mathcal{W}}; \mathbf{G}^*) && \mathbf{D}_w^* = \arg\min_{\mathbf{D}_w} \mathcal{L}_d(\mathbf{D}_w; \mathbf{G}^*), \forall w \in \mathcal{W} \end{aligned} \tag{12}$$

Fig. 2 depicts our generator and hierarchical discriminator. Pseudo-code for our algorithm can be found in Appendix B.

3.3 Inference Network

Once we have learned the counterfactual generator, we can use it only to access (generated) dose-response curves for all samples in the dataset. To generate dose-response curves for a new sample we use the counterfactual generator along with the original data to train an inference network, $\mathbf{I} : \mathcal{X} \times \mathcal{T} \to \mathcal{Y}$. As with the training of the generator and discriminator, we train using a random set of dosages, $\tilde{\mathcal{D}}_w$. The loss is given by

$$\mathcal{L}_I(\mathbf{I}) = \mathbb{E}\left[\sum_{w \in \mathcal{W}} \sum_{d \in \tilde{\mathcal{D}}_w} (\tilde{Y}(w, d) - \mathbf{I}(\mathbf{X}, (w, d)))^2 \right], \tag{13}$$

where $\tilde{Y}(w, d)$ is Y_f if $T_f = (w, d)$ or given by the generator if $T_f \neq (w, d)$. The expectation is taken over $\mathbf{X}, T_f, Y_f, \mathbf{Z}$ and $\tilde{\mathcal{D}}_w$.

4. Theoretical Analysis of SCIGAN

In this section we provide a theoretical analysis of the objective defined by Eq. (7) and establish equivalence with the game defined by Eqs. (12). Together, the results establish that our hierarchical GAN learns counterfactuals that agree (in marginal distribution) with the true data.

Lemma 1 *Fix* $\mathbf{G}$ *and* $\tilde{\mathcal{D}} = \bigcup_w \tilde{\mathcal{D}}_w$. *Let* $p_{w,d}(\mathbf{y}|\mathbf{x}) = p_r(y_{w,d}|\mathbf{x}) p_{\mathbf{G}}(\mathbf{y}_{\neg w,d}|\mathbf{x}, y_{w,d})$ *denote the induced joint density of outcomes when restricted to dosages in* $\tilde{\mathcal{D}}$, *where* p_r *denotes the true density that generated the observed outcome and* $p_{\mathbf{G}}$ *denotes the density induced by* $\mathbf{G}$ *over the remaining dosages in* $\tilde{\mathcal{D}}$. *Then the optimal discriminator is*

$$\mathbf{D}_{w,j}^*(\mathbf{x}, \mathbf{y}) = \frac{\tilde{p}(w, d_j|\mathbf{x}) p_{w,d_j}(\mathbf{y}|\mathbf{x})}{\sum_{w' \in \mathcal{W}} \sum_{i=1}^{n_w} \tilde{p}(w', d_i|\mathbf{x}) p_{w',d_i}(\mathbf{y}|\mathbf{x})} \tag{14}$$

where $\tilde{p}$ *is the* $\tilde{\mathcal{D}}$*-restricted propensity given by* $\tilde{p}(w, d_j|\mathbf{x}) = p(w|\mathbf{x})(p(d_j|\mathbf{x}, w) / \sum_{i=1}^{n_w} p(d_i|\mathbf{x}, w))$.

Proof Fix $\mathbf{G}$ and $\tilde{\mathcal{D}} = \bigcup_w \tilde{\mathcal{D}}_w$. The optimal discriminator is given by $\arg\min_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \mathbf{G})$. We have

$$\mathcal{L}(\mathbf{D}, \mathbf{G}) = \mathbb{E}\left[\sum_{w \in \mathcal{W}} \sum_{d \in \tilde{\mathcal{D}}_w} \mathbb{I}_{\{T_f = (w,d)\}} \log \mathbf{D}^{w,d}(\mathbf{X}, \tilde{\mathbf{Y}}) + \mathbb{I}_{\{T_f \neq (w,d)\}} \log(1 - \mathbf{D}^{w,d}(\mathbf{X}, \tilde{\mathbf{Y}})) \right] \tag{15}$$

$$\begin{aligned} &= \mathbb{E}_{\tilde{\mathcal{D}}}\left[\sum_{w \in \mathcal{W}} \sum_{d \in \tilde{\mathcal{D}}_w} \int_{(\mathbf{x},\mathbf{y})} \tilde{p}(w, d|\mathbf{x}) p_{w,d}(\mathbf{y}|\mathbf{x}) \log \mathbf{D}^{w,d}(\mathbf{x}, \mathbf{y}) \right. \\ &\qquad \left. + \left(\sum_{w',d' \neq w,d} \tilde{p}(w', d'|\mathbf{x}) p_{w',d'}(\mathbf{y}|\mathbf{x}) \right) \log(1 - \mathbf{D}^{w,d}(\mathbf{x}, \mathbf{y})) p(\mathbf{x}) d\mathbf{y} d\mathbf{x} \right] \end{aligned} \tag{16}$$

where we have taken the (conditional on $\tilde{\mathcal{D}}$) expectations inside the sums and replaced indicator functions with densities as appropriate. We now note that $a \log p + b \log(1-p)$ for $p \in (0,1)$ has a unique maximum at $p = \frac{a}{a+b}$, thus implying that the integrand is maximised when

$$\mathbf{D}^{w,d}(\mathbf{x}, \mathbf{y}) = \frac{\tilde{p}(w,d|\mathbf{x}) p_{w,d}(\mathbf{y}|\mathbf{x})}{\tilde{p}(w,d|\mathbf{x}) p_{w,d}(\mathbf{y}|\mathbf{x}) + \sum_{w',d' \neq w,d} \tilde{p}(w',d'|\mathbf{x}) p_{w',d'}(\mathbf{y}|\mathbf{x})} \,. \tag{17}$$

This gives the required result. ■

Using Lemma 1 we can now prove the key result, which establishes that the marginal distributions of our counterfactuals will be correct when estimated using our GAN framework. Importantly, this suffices for estimating μ since the expectation is only concerned with the marginal distribution of $Y(w,d)$.

Theorem 1 *The global minimum of the minimax game defined by* $\min_{\mathbf{G}} \max_{\mathbf{D}} \mathcal{L}(\mathbf{D}, \mathbf{G})$ *is achieved if and only if for all* $\tilde{\mathcal{D}}_w$*, for all* $w, w' \in \mathcal{W}$ *and for all* $d \in \tilde{\mathcal{D}}_w$*,* $d' \in \tilde{\mathcal{D}}_{w'}$

$$p_{w,d}(\mathbf{y}|\mathbf{x}) = p_{w',d'}(\mathbf{y}|\mathbf{x}) \tag{18}$$

which in turn implies that for any $(w,d) \in \mathcal{T}$ *we have that the generated counterfactual for outcome* (w,d) *for any sample (that was not assigned* (w,d)*) has the same (marginal) distribution (conditional on the features) as the true marginal distribution for that outcome.*

Proof For fixed $\tilde{\mathcal{D}}$ and $\mathbf{x}$ we note that by substituting the optimal discriminator into $\mathcal{L}(\mathbf{D}, \mathbf{G})$ and subtracting $\sum_{w \in \mathcal{W}} \sum_{i=1}^{n_w} \log \tilde{p}(w, d_i|\mathbf{x})$ (which is independent of $\mathbf{G}$) we obtain

$$\begin{aligned}\mathcal{L}(\mathbf{D}^*, \mathbf{G}) &- \int_{\mathbf{x}} \left(\sum_{w \in \mathcal{W}} \sum_{i=1}^{n_w} \log \tilde{p}(w, d_i|\mathbf{x}) \right) p(\mathbf{x}) d\mathbf{x} \\ &= \mathbb{E}_{\tilde{\mathcal{D}}} \int_{\mathbf{x}} \mathrm{KL}\Big(p_{w,d}(\mathbf{y}|\mathbf{x}) || \hat{p}(\mathbf{y}|\mathbf{x})\Big) + \mathrm{KL}\Big(\frac{1}{1 - \tilde{p}(w,d|\mathbf{x})} \sum_{t' \neq (w,d)} \tilde{p}(t'|\mathbf{x}) p_{t'}(\mathbf{y}|\mathbf{x}) || \hat{p}(\mathbf{y}|\mathbf{x})\Big) d\mathbf{x} \,. \end{aligned} \tag{19}$$

where KL is the KL divergence and $\hat{p}(\mathbf{y}|\mathbf{x}) = \sum_{t \in \tilde{\mathcal{T}}} \tilde{p}(t|\mathbf{x}) p_t(\mathbf{y}|\mathbf{x})$ where $\tilde{\mathcal{T}}$ is the restriction of $\mathcal{T}$ to the dosages in $\tilde{\mathcal{D}}$. We then note that that the KL divergence is minimised if and only if the two densities are equal, and we note by definition of $\hat{p}$ this occurs if and only if $p_{w,d}(\mathbf{y}|\mathbf{x}) = p_{w',d'}(\mathbf{y}|\mathbf{x})$ for all w, d, w', d'. This also directly implies that the marginal distributions for any fixed treatment-dosage pair agree for all factually observed treatments. In particular, if a sample received treatment $t' \neq t$, we have that the counterfactual generated for t for this sample has the same distribution as the true data generating distribution. ■

Our final theorem establishes equivalence between the hierarchical discriminator and the single discriminator setup under a mild assumption that can be satisfied by, say, resampling the noise for each treatment or passing independent noise samples to each multi-task head.

Theorem 2 *The game defined by Eqs. 12 is equivalent to the one defined by Eq. 7 if the response curves generated by the generator for different treatments are conditionally independent given the features.*

Proof To prove this result, it suffices to show that for fixed $\mathbf{G}$, $\mathbf{D}_H^* = \mathbf{D}^*$. To show this, we observe that by the same arguments as given for Lemma 1, we have the following:

$$\mathbf{D}_{\mathcal{W}}^{w\,*}(\mathbf{x}, \mathbf{y}) = \frac{p(w|\mathbf{x})\Big(\sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w) p_{w,d_i}(\mathbf{y}|\mathbf{x})\Big)}{\sum_{w' \in \mathcal{W}} \Big(p(w'|\mathbf{x}) \sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w)\Big)} \tag{20}$$

$$\mathbf{D}_w^{j\,*}(\mathbf{x}, \mathbf{y}_w) = \frac{\tilde{p}(d_j|\mathbf{x}, w) p_{w,d_j}(\mathbf{y}_w|\mathbf{x})}{\sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w) p_{w,d_i}(\mathbf{y}_w|\mathbf{x})} \tag{21}$$

where $\mathbf{y}_w$ is the restriction of $\mathbf{y}$ to the outcomes corresponding to treatment w. By multiplying (21) by $\frac{p_{w,d_j}(\mathbf{y}_{\neq w}|\mathbf{y},\mathbf{x})}{p_{w,d_j}(\mathbf{y}_{\neq w}|\mathbf{y},\mathbf{x})}$ we obtain

$$\mathbf{D}_w^{j\,*}(\mathbf{x}, \mathbf{y}_w) = \frac{\tilde{p}(d_j|\mathbf{x}, w) p_{w,d_j}(\mathbf{y}|\mathbf{x})}{\sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w) p_{w,d_i}(\mathbf{y}|\mathbf{x})} \tag{22}$$

since the conditional independence assumption implies that $p_{w,d_j}(\mathbf{y}_{\neq w}|\mathbf{y}, \mathbf{x}) = p_{w,d_i}(\mathbf{y}_{\neq w}|\mathbf{y}, \mathbf{x})$ for all $i, j = 1, ..., n_w$. Multiplying (20) and (22) together to get $\mathbf{D}_H^{w,j}$, we notice that the denominator in (22) cancels with the bracketed term of the numerator in (20) to give

$$\mathbf{D}_H^{*\,w,j} = \frac{p(w|\mathbf{x})\tilde{p}(d_j|\mathbf{x}, w) p_{w,d_j}(\mathbf{y}|\mathbf{x})}{\sum_{w' \in \mathcal{W}} \Big(p(w'|\mathbf{x}) \sum_{i=1}^{n_w} \tilde{p}(d_i|\mathbf{x}, w)\Big)} \tag{23}$$

$$= \frac{\tilde{p}(w, d_j|\mathbf{x}) p_{w,d_j}(\mathbf{y}|\mathbf{x})}{\sum_{w' \in \mathcal{W}} \sum_{i=1}^{n_w} \tilde{p}(w', d_i|\mathbf{x}) p_{w',d_i}(\mathbf{y}|\mathbf{x})} \tag{24}$$

which is equal to the optimal discriminator for the single loss given in Lemma 1. ■

5. Architecture

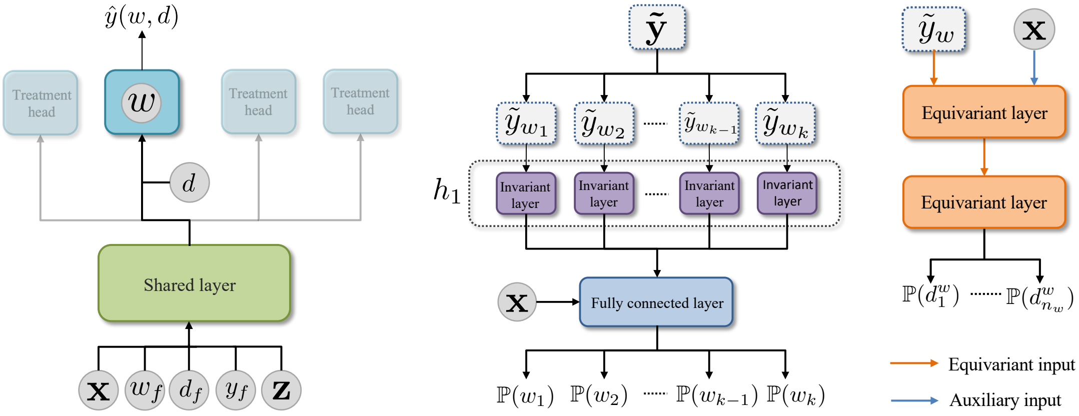
In this section, we describe in detail the novel architectures that we adopt to model each of the functions $\mathbf{G}, \mathbf{D}, \mathbf{D}_{\mathcal{W}}, \mathbf{D}_{w_1}, ..., \mathbf{D}_{w_k}$ which draws from the ideas in Zaheer et al. (2017). The inference network, $\mathbf{I}$, has the same architecture as the generator, but does not receive w_f, d_f, y_f or $\mathbf{z}$ as inputs.

5.1 Generator Architecture

We adopt a multi-task deep learning model for $\mathbf{G}$ by defining a function $g : \mathcal{X} \times \mathcal{T} \times \mathcal{Y} \times \mathcal{Z} \rightarrow \mathcal{H}$ for some latent space $\mathcal{H}$ (typically $\mathbb{R}^l$ for some l) and then for each treatment $w \in \mathcal{W}$ we introduce a multitask "head", $g_w : \mathcal{H} \times \mathcal{D}_w \rightarrow \mathcal{Y}$ taking inputs from $\mathcal{H}$ *and* a dosage, d, to produce an outcome $\hat{y}(w, d) \in \mathcal{Y}$. Given observations, $(\mathbf{x}, t_f, y_f)$, a noise vector $\mathbf{z}$, and a target treatment-dosage pair, $t = (w, d)$, we define

$$\mathbf{G}(\mathbf{x}, t_f, y_f, \mathbf{z})(t) = g_w(g(\mathbf{x}, t_f, y_f, \mathbf{z}), d)\,. \tag{25}$$

Each of $g, g_{w_1}, ..., g_{w_k}$ are fully connected networks. A figure of our generator architecture is given in Figure 3(a).



(a) Generator

(b) Treatment Discriminator

(c) Dosage Discriminator

Figure 3: Architecture of our generator and discriminators.

5.2 Discriminator Architectures

As noted in Section 1, our discriminators need to act as functions of sets (of randomly selected dosage-outcome pairs). While we could require that our discriminators try to learn this during training, by enforcing them to be functions of sets through their architecture, we reduce the complexity of learning the discriminators (they no longer need to “rule out" functions which are not functions of sets). This results in better performing discriminators, which in turn improves the performance of the generator.

In practice, the treatment discriminator receives all of the sets (i.e. one set for each treatment) of dosage-outcome pairs and outputs a probability for each treatment (i.e. there is one output corresponding to each set). In order to define such a function, we treat each input set as a vector but require that the outputs be invariant to (i.e. should not depend on) the ordering of the set as a vector. Each dosage discriminator receives the set corresponding to a given treatment and is tasked with outputting a probability for each element in the set. In order to define such a function, we consider the input and output as vectors but then require that if we permute the elements of the input vector, the output should be permuted in the same way.

5.2.1 Permutation Invariance and Permutation Equivariance

The notions of what it means for a function to be *permutation invariant* and *permutation equivariant* with respect to (a subset of) its inputs are given below in definitions 1 and 2, respectively. Let $\mathcal{U}$, $\mathcal{V}$, $\mathcal{C}$ be some spaces. Let $m \in \mathbb{Z}^+$.

Definition 1 *A function $f : \mathcal{U}^m \times \mathcal{V} \to \mathcal{C}$ is permutation* invariant *with respect to the space $\mathcal{U}^m$ if for every $\mathbf{u} = (u_1, ..., u_m) \in \mathcal{U}^m$, every $v \in \mathcal{V}$ and every permutation, σ, of $\{1, ..., m\}$ we have*

$$f(u_1, ..., u_m, v) = f(u_{\sigma(1)}, ..., u_{\sigma(m)}, v)\,. \tag{26}$$

Definition 2 *A function $f : \mathcal{U}^m \times \mathcal{V} \to \mathcal{C}^m$ is permutation* equivariant *with respect to the space $\mathcal{U}^m$ if for every $\mathbf{u} \in \mathcal{U}^m$, every $v \in \mathcal{V}$ and every permutation, σ, of $\{1, ..., m\}$ we have $f(u_{\sigma(1)}, ..., u_{\sigma(m)}, v) = (f_{\sigma(1)}(\mathbf{u}, v), ..., f_{\sigma(m)}(\mathbf{u}, v))$, where $f_j(\mathbf{u}, v)$ is the jth element of $f(\mathbf{u}, v)$.*

To build up functions that are permutation invariant and permutation equivariant we make the following observations: (1) the composition of any function with a permutation invariant function is permutation invariant, (2) the composition of two permutation equivariant functions is permutation equivariant.

Zaheer et al. (2017) provide several possible building blocks to use to construct invariant and equivariant deep networks. The basic building block we will use for invariant functions will be a layer of the form

$$f_{inv}(\mathbf{u}) = \sigma(\mathbf{1}_b \mathbf{1}_m^T(\phi(u_1), ..., \phi(u_m)))\,, \tag{27}$$

where $\mathbf{1}_l$ is a vector of 1s of dimension l, ϕ is any function $\phi : \mathcal{U} \to \mathbb{R}^q$ for some q (in this paper we use a standard fully connected layer) and σ is some non-linearity.

The basic building block for equivariant functions is defined in terms of equivariance input, $\mathbf{u}$, and auxiliary input, $\mathbf{v}$, by

$$f_{equi}(\mathbf{u}, \mathbf{v}) = \sigma(\lambda \mathbf{I}_m \mathbf{u} + \gamma(\mathbf{1}_m \mathbf{1}_m^T)\mathbf{u} + (\mathbf{1}_m \Theta^T)\mathbf{v})\,, \tag{28}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix, λ and γ are scalar parameters and Θ is a vector of weights.

5.2.2 Hierarchical Discriminator Architecture

In the case of the hierarchical discriminator, we want the treatment discriminator, $\mathbf{D}_{\mathcal{W}}$, to be permutation invariant with respect to $\tilde{\mathbf{y}}_w$ for each treatment. To achieve this we define $h_1 : \prod_{w \in \mathcal{W}} (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to \mathcal{H}_H$ and require that h_1 be permutation invariant w.r.t. each of the spaces $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$. We concatenate the output of h_1 with the features $\mathbf{x}$ and pass these through a fully connected network $h_2 : \mathcal{X} \times \mathcal{H}_H \to [0, 1]^k$ so that $\mathbf{D}_{\mathcal{W}}(\mathbf{x}, \tilde{\mathbf{y}}) = h_2(\mathbf{x}, h_1(\tilde{\mathbf{y}}))$.

To construct h_1, we concatenate the outputs of several invariant layers of the form given in Eq. (27) that each individually act on the spaces $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$. That is, for each treatment, $w \in \mathcal{W}$ we define a map $h_{inv}^w : (\mathcal{D}_w \times \mathcal{Y})^{n_w} \to \mathcal{H}_H^w$ by substituting $\tilde{\mathbf{y}}_w$ for $\mathbf{u}$ in Eq. (27). We then define $\mathcal{H}_H = \prod_{w \in \mathcal{W}} \mathcal{H}_H^w$ and $h_1(\tilde{\mathbf{y}}) = (h_{inv}^{w_1}(\tilde{\mathbf{y}}_{w_1}), ..., h_{inv}^{w_k}(\tilde{\mathbf{y}}_{w_k}))$.

We want each dosage discriminator, $\mathbf{D}_w$, to be permutation equivariant with respect to $\tilde{\mathbf{y}}_w$. To achieve this each $\mathbf{D}_w$ will consist of two layers of the form given in Eq. (28) with the equivariance input, $\mathbf{u}$, to the first layer being $\tilde{\mathbf{y}}_w$ and to the second layer being the output of the first layer and the auxiliary input, $\mathbf{v}$, to the first layer being the features, $\mathbf{x}$, and then no auxiliary input to the second layer. Diagrams depicting the architectures of the treatment discriminator and dosage discriminators can be found in Fig. 3(b) and Fig. 3(c) respectively.

6. Related work

Methods for estimating the outcomes of treatments with a continuous dosage from observational data make use of the generalized propensity score (GPS) (Imbens, 2000; Imai and Van Dyk, 2004; Hirano and Imbens, 2004) or build on top of balancing methods for multiple treatments. Schwab et al. (2019) developed a neural network based method to estimate counterfactuals for multiple treatments and continuous dosages. The proposed Dose Response networks (DRNets) in Schwab et al. (2019) consist of a three level architecture with shared layers for all treatments, multi-task layers for each treatment and additional multi-task layers for dosage sub-intervals. Specifically, for each treatment w, the dosage interval $[a_w, b_w]$ is subdivided into E *equally* sized sub-intervals and a multi-task head is added for each sub-interval. This is an extension of the architecture in Shalit et al. (2017). However, the main advantage of using multi-task heads for dosage intervals would be the added flexibility in the model to learn potentially very different functions over different regions of the dosage interval. DRNets do not determine these intervals dynamically and thus much of this flexbility is lost. Our approach (using GANs to generate counterfactuals) fundamentally differs from DRNets (supervised learning with bias-adjustment) and we demonstrate experimentally that SCIGAN outperforms both GPS and DRNets.

Most methods for performing causal inference in the static setting focus on the scenario with two or multiple treatment options and no dosage parameter. The approaches taken by such methods to estimate the treatment effects involve either building a separate regression model for each treatment (Stoehlmacher et al., 2004; Qian and Murphy, 2011; Bertsimas et al., 2017) or using the treatment as a feature and adjusting for the imbalance between the different treatment populations. The former does not generalise to the dosage setting due to the now infinite number of possible treatments available. Note also that treating the dosage as an input does not account for the bias in the dosage parameter. In the latter case, methods for handling the selection bias involve propensity weighting (Crump et al., 2008; Alaa et al., 2017; Shi et al., 2019), building sub-populations using tree based methods (Chipman et al., 2010; Athey and Imbens, 2016; Wager and Athey, 2018; Kallus, 2017) or building balancing representations between patients receiving the different treatments (Johansson et al., 2016; Shalit et al., 2017; Li and Fu, 2017; Yao et al., 2018). An additional approach involves modelling the data distribution of the factual and counterfactual outcomes (Alaa and van der Schaar, 2017; Yoon et al., 2018).

Silva (2016) leverages observational and interventional data to estimate the effects of discrete dosages for a single treatment. In particular, Silva (2016) uses observational data to construct a non-stationary covariance function and develop a hierarchical Gaussian process prior to build a distribution over the dose response curve. Then, controlled interventions are employed to learn a non-parametric affine transform to reshape this distribution. The setting in Silva (2016) differs significantly from ours as we do not assume access to any interventional data.

7. Evaluation

The nature of the treatment-effects estimation problem in even the binary treatments setting does not allow for meaningful evaluation on real-world datasets due to the inability to observe the counterfactuals. While there are well-established benchmark synthetic models for use in the binary (or multiple) case, no such models exist for the dosage setting. We propose our own semi-synthetic data simulation to evaluate our model against several benchmarks.

7.1 Experimental setup

7.1.1 Semi-synthetic data generation

We simulate data as follows. We obtain features, $\mathbf{x}$, from a real dataset (in this paper we use TCGA (Weinstein et al., 2013), News (Johansson et al., 2016; Schwab et al., 2019)) and MIMIC III (Johnson et al., 2016))[4]. We consider 3 treatments each accompanied by a dosage. Each treatment, w, is associated with a set of parameters, $\mathbf{v}_1^w, \mathbf{v}_2^w, \mathbf{v}_3^w$. For each run of the experiment, these parameters are sampled randomly by sampling a vector, $\mathbf{u}_i^w$, from $\mathcal{N}(\mathbf{0}, \mathbf{1})$ and then setting $\mathbf{v}_i^w = \mathbf{u}_i^w / ||\mathbf{u}_i^w||$ where $|| \cdot ||$ is Euclidean norm. The shape of the response curve for each treatment, $f_w(\mathbf{x}, d)$, is given in Table 1, along with a closed-form expression for the optimal dosage. We add $\epsilon \sim \mathcal{N}(0, 0.2)$ noise to the outcomes.

We assign interventions by sampling a dosage, d_w, for each treatment from a beta distribution

$$d_w | \mathbf{x} \sim \text{Beta}(\alpha, \beta_w) \,. \tag{29}$$

$\alpha \geq 1$ controls the selection bias. We set $\beta_w = \frac{\alpha - 1}{d_w^*} + 2 - \alpha$ where d_w^* is the optimal dosage for treatment w (which is a function of $\mathbf{x}$). This setting of β_w ensures that the mode of our distribution is d_w^*, and we can

[4] Details of each dataset can be found in Appendix E

write the variance of d_w in terms of α and d_w^* as follows

$$\mathrm{Var}(d_w) = \frac{\frac{\alpha^2-\alpha}{d_w^*} + 2\alpha - \alpha^2}{(\frac{\alpha-1}{d_w^*}+2)^2(\frac{\alpha-1}{d_w^*}+3)} \approx \frac{c\alpha^2}{d\alpha^3}\,. \tag{30}$$

We see that the variance of our Beta distribution therefore decreases with α, resulting in the sampled dosages being closer to the optimal dosage, thus resulting in higher dosage-selection bias. In addition we note that the $\mathrm{Beta}(1, 1)$ distribution is the uniform distribution, corresponding to the dosages being sampled independently of the patient features, resulting in no selection bias when $\alpha = 1$. Note that when $d_w^* = 0$, for symmetry, we sample d_w from $1 - \mathrm{Beta}(\alpha, \beta_w)$ where β_w is set as though $d_w^* = 1$.

Given a selected dosage for each treatment, we assign a treatment according to

$$w_f|\mathbf{x} \sim \mathrm{Categorical}(\mathrm{softmax}(\kappa f(\mathbf{x}, d_w)) \tag{31}$$

where increasing κ increases selection bias, and $\kappa = 0$ leads to random assignments. The factual intervention is given by (w_f, d_{w_f}). Unless otherwise specified, we set $\kappa = 2$ and $\alpha = 2$.

We consider 3 shapes for f_w to demonstrate learning heterogeneous response curves. The first curve can be broken down into two terms, a linear (in d) increasing term $(\mathbf{v}_1^1)^T\mathbf{x} + 12(\mathbf{v}_2^1)^T\mathbf{x}d$ and a quadratic (in d) decreasing term $-12(\mathbf{v}_3^1)^T\mathbf{x}d^2$. This first term could represent the improved efficacy of higher dosages of chemotherapy in reducing tumour size, while the quadratic term could represent the increasing toxicity of chemotherapy as the dosage increases. This type of trade-off presents itself in many other settings.

Treatment	Dose-Response	Optimal dosage
1	$f_1(\mathbf{x}, d) = C((\mathbf{v}_1^1)^T\mathbf{x} + 12(\mathbf{v}_2^1)^T\mathbf{x}d - 12(\mathbf{v}_3^1)^T\mathbf{x}d^2)$	$d_1^* = \frac{(\mathbf{v}_2^1)^T\mathbf{x}}{2(\mathbf{v}_3^1)^T\mathbf{x}}$
2	$f_2(\mathbf{x}, d) = C((\mathbf{v}_1^2)^T\mathbf{x} + \sin(\pi(\frac{\mathbf{v}_2^{2\,T}\mathbf{x}}{\mathbf{v}_3^{2\,T}\mathbf{x}})d))$	$d_2^* = \frac{(\mathbf{v}_3^2)^T\mathbf{x}}{2(\mathbf{v}_2^2)^T\mathbf{x}}$
3	$f_3(\mathbf{x}, d) = C((\mathbf{v}_1^3)^T\mathbf{x} + 12d(d-b)^2$, where $b = 0.75\frac{(\mathbf{v}_2^3)^T\mathbf{x}}{(\mathbf{v}_3^3)^T\mathbf{x}})$	$\frac{b}{3}$ if $b \geq 0.75$ 1 if $b < 0.75$

Table 1: Dose response curves used to generate semi-synthetic outcomes for patient features $\mathbf{x}$. In the experiments, we set $C = 10$. $\mathbf{v}_1^w, \mathbf{v}_2^w, \mathbf{v}_3^w$ are the parameters associated with each treatment w.

7.1.2 Benchmarks

We compare against two benchmarks: Generalized Propensity Score (GPS) (Imbens, 2000) and Dose Reponse Networks (DRNet) (Schwab et al., 2019) (the standard model and with Wasserstein regularization (DRN-W)). As a baseline, we compare against a standard multilayer perceptron (MLP) that takes patient features, treatment and dosage as input and estimates the patient outcome and a multitask variant (MLP-M) that has a designated head for each treatment. See Appendix F for details of the benchmark models and their hyperparameter optimisation.

7.1.3 Metrics

For metrics, we use Mean Integrated Square Error (MISE), Dosage Policy Error (DPE) and Policy Error (PE) (Silva, 2016; Schwab et al., 2019). Each of these metrics are computed on a held out test-set.

The Mean Integrated Square Error (MISE) measures how well a model estimates the patient outcomes across the entire dosage space:

$$\text{MISE} = \frac{1}{N}\frac{1}{k}\sum_{w\in\mathcal{W}}\sum_{i=1}^{N}\int_{\mathcal{D}_w}\Big(y^i(w,u) - \hat{y}^i(w,u)\Big)^2 \mathrm{d}u\,. \tag{32}$$

The mean dosage policy error (DPE) (Schwab et al., 2019) assesses the ability of a model to estimate the optimal dosage point for every treatment for each individual:

$$\text{DPE} = \frac{1}{N}\frac{1}{k}\sum_{w\in\mathcal{W}}\sum_{i=1}^{N}\Big(y^i(w,d_w^*) - y^i(w,\hat{d}_w^*)\Big)^2\,, \tag{33}$$

where d_w^* is the true optimal dosage and $\hat{d}_w^*$ is the optimal dosage identified by the model. The optimal dosage points for a model are computed using SciPy's implementation of Sequential Least SQuares Programming.

Finally, the mean policy error (PE) (Schwab et al., 2019) compares the outcome of the true optimal treatment-dosage pair to the outcome of the optimal treatment-dosage pair as selected by the model:

$$\text{PE} = \frac{1}{N}\sum_{i=1}^{N}\Big(y^i(w^*,d_w^*) - y^i(\hat{w}^*,\hat{d}_w^*)\Big)^2\,, \tag{34}$$

where w^* is the true optimal treatment and $\hat{w}^*$ is the optimal treatment identified by the model. The optimal treatment-dosage pair for a model is selected by first computing the optimal dosage for each treatment and then selecting the treatment with the best outcome for its optimal dosage.

7.2 Source of gain

Before comparing against the benchmarks, we investigate how each component of our model affects performance. We start with a baseline model in which both the generator and discriminator consist of a single fully connected network. One at a time, we add in the following components (cumulatively until we reach our full model): (1) the supervised loss in Eq. 4 (+ $\mathcal{L}_S$), (2) multitask heads in the generator (+ Multitask), (3) hierarchical discriminator (+ Hierarchical) and (4) invariance/equivariance layers in the treatment and dosage discriminators (+Inv/Eqv). We report the results in Table 2 for TCGA and News for all 3 error metrics (MISE, DPE and PE), computed over 30 runs.

		TCGA			**News**	
	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$
Baseline	4.18 ± 0.32	2.06 ± 0.16	1.93 ± 0.12	6.17 ± 0.27	6.97 ± 0.27	6.20 ± 0.21
+ $\mathcal{L}_S$	3.37 ± 0.11	1.14 ± 0.05	0.84 ± 0.05	4.51 ± 0.16	4.46 ± 0.12	4.40 ± 0.11
+ Multitask	3.15 ± 0.12	0.85 ± 0.05	0.67 ± 0.05	4.11 ± 0.11	4.33 ± 0.11	4.31 ± 0.11
+ Hierarchical	2.54 ± 0.05	0.36 ± 0.05	0.45 ± 0.05	4.07 ± 0.05	4.24 ± 0.11	4.17 ± 0.12
+ Inv/Eqv	1.89 ± 0.05	0.31 ± 0.05	0.25 ± 0.05	3.71 ± 0.05	4.14 ± 0.11	3.90 ± 0.05

Table 2: Source of gain analysis for our model. Metrics are reported as Mean ± Std.

The addition of each component results in improved performance, with the final row (our full model) demonstrating the best performance across both datasets and for all metrics. In Appendix G we further compare our hierarchical discriminator with a single network discriminator by investigating both models sensitivity to the hyperparameter n_w. Details of the single discriminator can be found in Appendix D.

7.3 Benchmarks comparison

We now compare SCIGAN against the benchmarks on our 3 semi-synthetic datasets. For MIMIC, due to the low number of samples available, we use two treatments - 2 and 3. We report each metric in Table 3. We see that SCIGAN demonstrates a significant improvement over every benchmark across all 3 datasets. In section 7.6 we compare SCIGAN with DRNET and GPS for an increasing number of treatments.

Method		**TCGA**			**News**			**MIMIC**	
	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$
SCIGAN	$\mathbf{1.89 \pm 0.05}$	$\mathbf{0.31 \pm 0.05}$	$\mathbf{0.25 \pm 0.05}$	$\mathbf{3.71 \pm 0.05}$	$\mathbf{4.14 \pm 0.11}$	$\mathbf{3.90 \pm 0.05}$	$\mathbf{2.09 \pm 0.12}$	$\mathbf{0.51 \pm 0.05}$	$\mathbf{0.32 \pm 0.05}$
DRNet	3.64 ± 0.12	0.51 ± 0.05	0.67 ± 0.05	4.98 ± 0.12	4.39 ± 0.11	4.17 ± 0.11	4.45 ± 0.12	0.52 ± 0.05	1.44 ± 0.05
DRN-W	3.71 ± 0.12	0.50 ± 0.05	0.63 ± 0.05	5.07 ± 0.12	4.21 ± 0.11	4.56 ± 0.12	4.47 ± 0.12	0.53 ± 0.05	1.37 ± 0.05
GPS	4.83 ± 0.01	1.38 ± 0.01	1.60 ± 0.01	6.97 ± 0.01	6.40 ± 0.01	24.1 ± 0.05	7.39 ± 0.00	1.41 ± 0.12	20.2 ± 0.01
MLP-M	3.96 ± 0.12	0.92 ± 0.05	1.20 ± 0.05	5.17 ± 0.12	4.94 ± 0.16	5.82 ± 0.16	4.97 ± 0.16	0.77 ± 0.05	1.59 ± 0.05
MLP	4.31 ± 0.05	1.04 ± 0.05	0.97 ± 0.05	5.48 ± 0.16	5.18 ± 0.12	6.45 ± 0.21	5.34 ± 0.16	0.80 ± 0.05	1.65 ± 0.05

Table 3: Performance of individualized treatment-dose response estimation on three datasets. Bold indicates the method with the best performance for each dataset. Metrics are reported as Mean ± Std.

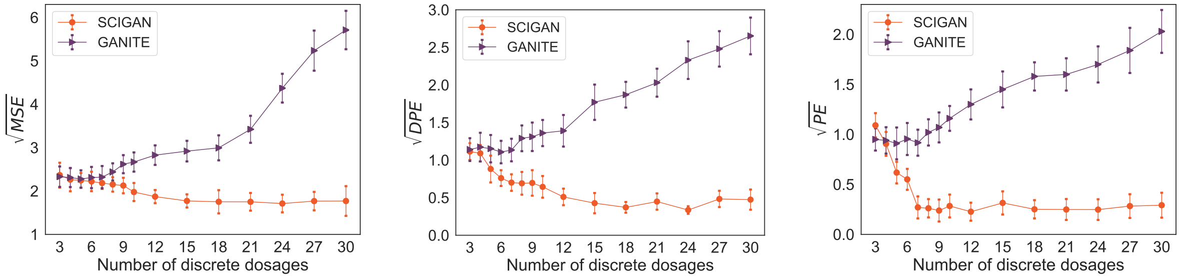
7.4 Discrete dosages

In this experiment, we investigate the discrete dosage setting. In this set-up, we use the TCGA dataset and treatments 2 and 3 from Table 1 with dose-response curves $f_2(\mathbf{x}, d)$ and $f_3(\mathbf{x}, d)$ respectively. Let β be the number of discrete dosages for which we want to generate data. We chose β equally spaced points in the interval $[0, 1]$ as our set of discrete dosages: $\Delta = \{\frac{k}{\beta-1}\}_{k=0}^{\beta-1}$. To create factual dosages for our dataset, we sample the dosages as before $d_w \mid x \sim \text{Beta}(\alpha, \beta_w)$, and choose the closest discrete dosage from the set Δ.

To evaluate SCIGAN in this setting, we maintain the same architecture for the multi-task generator and hierarchical discriminator. The only difference is that we now randomly sample dosages for the SCIGAN discriminator from Δ.

We adopt the GANITE implementation proposed by Yoon et al. (2018). To be able to have a fair comparison with SCIGAN we also use a multi-task architecture for the GANITE generator and we give as input to each multitask head the dosage parameter. The GANITE generator will generate outcomes for all possible discrete dosages in Δ and these will be passed to the GANITE discriminator to distinguish the factual one. For the GANITE generator we use a similar architecture to the SCIGAN generator with 2 hidden layers for each multitask head and 64 neurons in each layer. The GANITE discriminator consists of 2 fully connected layers with 64 neurons in each. We also set $\lambda = 1$. In addition, to maintain a similar set-up to SCIGAN, we train an inference network to learn the counterfactual outcomes with data from the GANITE generator. The inference network has the same architecture as the GANITE generator. We report the Mean Squared Error (MSE), DPE and PE of SCIGAN and GANITE in Fig. 4 where we vary the number of discrete dosages from 3 to 30.

We clearly see from Fig. 4 that SCIGAN achieves a similar performance to GANITE for a small number of dosages (< 6) but then significantly outperforms GANITE for more dosages than 6. In fact,



(a) $\sqrt{\text{MSE}}$ (b) $\sqrt{\text{DPE}}$ (c) $\sqrt{\text{PE}}$

Figure 4: Comparison between SCIGAN and GANITE in the discrete dosage set-up.

we see that while GANITE's performance degrades with an increasing number of dosages, SCIGAN's improves and then stabilises at around 12 dosages. This is due to the fact that the single discriminator in GANITE simply cannot handle a large number of dosages. Our hierarchical model, however, can. The worse performance of SCIGAN for the lower dosages can be attributed to the fact that for such few dosages (e.g. 3 dosages corresponds to only 6 total different interventions), the SCIGAN architecture is overly complex, and the sub-sampling of dosages for the discriminator is not actually necessary. These results demonstrate SCIGAN's wide-ranging applicability in both discrete and continuous settings.

7.5 Mixing dosage and no-dosage treatment options

We also evaluate the case when one of the treatments does not have a dosage parameter. For this experiment we generate data for the treatment that has a dosage parameter d using $f_3(\mathbf{x}, d)$ and for the treatment without an associated dosage using $2C(\mathbf{v}_0^T\mathbf{x})$, where $\mathbf{v}_0$ are parameters, $\mathbf{x}$ are patient features and $C =$ is the scaling parameter. This set-up also corresponds to the scenario where we want to compare giving a treatment with a dosage and not giving any treatment.

SCIGAN can be easily extended to incorporate an additional treatment that does not come with a dosage parameter. Such treatments will not need a dosage discriminator but will be passed to the treatment discriminator. A head can be added to the generator for each such non-dosage treatment but will not need to take dosage as an input. As the DRNet public implementation does not allow for this set-up, we compared SCIGAN with the multilayer perceptron model with multitask heads (MLP-M). This model is trained using supervised learning to minimize error on the factual outcomes and consists of two multitask heads: one head for the treatment option which receives as input the dosage and estimates the dose-response curve and one head for the no-treatment option.

As can be seen in Table 4, SCIGAN is capable of handling this setting and lends itself naturally to potentially mixed dosage and no-dosage treatment options.

Method		**TCGA**			**News**			**MIMIC**	
	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$
SCIGAN	$\mathbf{1.28 \pm 0.09}$	$\mathbf{1.37 \pm 0.07}$	$\mathbf{1.56 \pm 0.06}$	$\mathbf{3.18 \pm 0.15}$	$\mathbf{2.04 \pm 0.09}$	$\mathbf{2.49 \pm 0.12}$	$\mathbf{0.61 \pm 0.08}$	$\mathbf{1.82 \pm 0.02}$	$\mathbf{1.89 \pm 0.03}$
MLP-M	2.08 ± 0.12	1.85 ± 0.16	2.02 ± 0.07	4.68 ± 0.11	2.45 ± 0.08	2.64 ± 0.08	1.56 ± 0.08	2.04 ± 0.03	2.14 ± 0.5

Table 4: Performance of individualized treatment-dose response when mixing treatment with no-treatment options. Bold indicates the method with the best performance for each dataset.

7.6 Varying the number of treatments

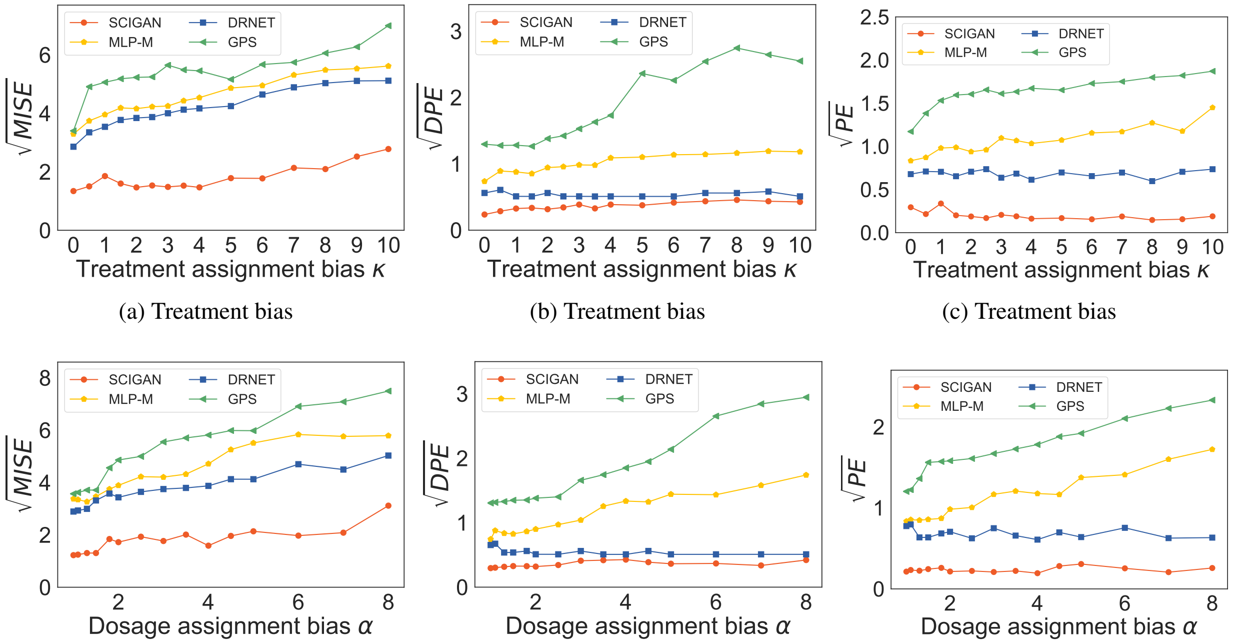
In this experiment, we increase the number of treatments by defining 3 or 6 additional treatments. The parameters $\mathbf{v}_1^w, \mathbf{v}_2^w, \mathbf{v}_3^w$ are defined in exactly the same way as for 3 treatments. The outcome shapes for treatments 4 and 7 are the same as for treatment 1, similarly for 5, 8 and 2 and for 6, 9 and 3. In Table 5 we report MISE, DPE and PE on the TCGA dataset with 6 treatments (TCGA-6) and with 9 treatments (TCGA-9). Note that we use 3 dosage samples for training SCIGAN in this experiment.

Method		TCGA - 6			TCGA - 9	
	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$	$\sqrt{\text{MISE}}$	$\sqrt{\text{DPE}}$	$\sqrt{\text{PE}}$
SCIGAN	$\mathbf{2.37 \pm 0.12}$	$\mathbf{0.43 \pm 0.05}$	$\mathbf{0.32 \pm 0.05}$	$\mathbf{2.79 \pm 0.05}$	$\mathbf{0.51 \pm 0.05}$	$\mathbf{0.54 \pm 0.05}$
DRNET	4.09 ± 0.16	0.52 ± 0.05	0.71 ± 0.05	4.31 ± 0.12	0.59 ± 0.05	0.74 ± 0.05
GPS	6.62 ± 0.01	2.04 ± 0.01	2.61 ± 0.00	7.58 ± 0.01	3.14 ± 0.01	2.91 ± 0.01

Table 5: Performance of SCIGAN and the benchmarks when we increase the number of treatments in the dataset to 6 and 9. Bold indicates the method with the best performance for each dataset.

7.7 Treatment and dosage selection bias

Finally, we assess each model's robustness to treatment and dosage bias. Fig. 5(a) shows the performance of the 4 methods for κ between 0 (no bias) and 10 (strong bias). Fig. 5(b) shows the performance for α between 1 (no bias) and 8 (strong bias). SCIGAN shows consistent performance, significantly outperforming the benchmarks for all κ and α.



(a) Treatment bias

(b) Treatment bias

(c) Treatment bias

(d) Dosage bias

(e) Dosage bias

(f) Dosage bias

Figure 5: Performance of the 4 methods on datasets with varying bias levels.

8. Conclusion

In this paper we proposed a novel framework for estimating response curves for continuous interventions from observational data. We provided theoretical justification for our use of a modified GAN framework, which introduced a novel hierarchical discriminator. We also proposed novel architectures and introduced a new semi-synthetic data simulation for use as a benchmark. On this data we demonstrated significant improvements over the benchmarks.

Acknowledgements

The research presented in this paper was supported by The Alan Turing Institute, under the EPSRC grant EP/N510129/1.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Ahmed M Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Advances in Neural Information Processing Systems*, pages 3424–3432, 2017.

Ahmed M Alaa, Michael Weisz, and Mihaela Van Der Schaar. Deep counterfactual networks with propensity-dropout. *arXiv preprint arXiv:1706.05966*, 2017.

Susan Athey and Guido Imbens. Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360, 2016.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

Dimitris Bertsimas, Nathan Kallus, Alexander M Weinstein, and Ying Daisy Zhuo. Personalized diabetes management using electronic medical records. *Diabetes Care*, 40(2):210–217, 2017.

Hugh A Chipman, Edward I George, Robert E McCulloch, et al. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

Richard K Crump, V Joseph Hotz, Guido W Imbens, and Oscar A Mitnik. Nonparametric tests for treatment effect heterogeneity. *The Review of Economics and Statistics*, 90(3):389–405, 2008.

Donna Döpp-Zemel and AB Johan Groeneveld. High-dose norepinephrine treatment: determinants of mortality and futility in critically ill patients. *American Journal of Critical Care*, 22(1):22–32, 2013.

Douglas Galagate. *Causal Inference with a Continuous Treatment and Outcome: Alternative Estimators for Parametric Dose-Response function with Applications.* PhD thesis, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

Keisuke Hirano and Guido W Imbens. The propensity score with continuous treatments. *Applied Bayesian Modeling and Causal Inference from Incomplete-Data Perspectives*, 226164:73–84, 2004.

Kosuke Imai and David A Van Dyk. Causal inference with general treatment regimes: Generalizing the propensity score. *Journal of the American Statistical Association*, 99(467):854–866, 2004.

Guido W Imbens. The role of the propensity score in estimating dose-response functions. *Biometrika*, 87 (3):706–710, 2000.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning*, pages 3020–3029, 2016.

Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific Data*, 3:160035, 2016.

Nathan Kallus. Recursive partitioning for personalization using observational data. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1789–1798. JMLR. org, 2017.

Sheng Li and Yun Fu. Matching on balanced nonlinear representations for treatment effects estimation. In *Advances in Neural Information Processing Systems*, pages 929–939, 2017.

Robert W Platt, Enrique F Schisterman, and Stephen R Cole. Time-modified confounding. *American journal of epidemiology*, 170(6):687–694, 2009.

Min Qian and Susan A Murphy. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180, 2011.

Donald B Rubin. Bayesianly justifiable and relevant frequency calculations for the applies statistician. *The Annals of Statistics*, pages 1151–1172, 1984.

Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M Buhmann, and Walter Karlen. Learning counterfactual representations for estimating individual dose-response curves. *arXiv preprint arXiv:1902.00981*, 2019.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3076–3085. JMLR. org, 2017.

Claudia Shi, David M Blei, and Victor Veitch. Adapting neural networks for the estimation of treatment effects. *arXiv preprint arXiv:1906.02120*, 2019.

Ricardo Silva. Observational-interventional priors for dose-response learning. In *Advances in Neural Information Processing Systems*, pages 1561–1569, 2016.

Peter Spirtes. A tutorial on causal inference. 2009.

J Stoehlmacher, DJ Park, W Zhang, D Yang, S Groshen, S Zahedy, and HJ Lenz. A multivariate analysis of genomic polymorphisms: Prediction of clinical outcome to 5-FU/Oxaliplatin combination chemotherapy in refractory colorectal cancer. *British Journal of Cancer*, 91(2):344, 2004.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

Kyle Wang, Michael J Eblan, Allison M Deal, Matthew Lipner, Timothy M Zagar, Yue Wang, Panayiotis Mavroidis, Carrie B Lee, Brian C Jensen, Julian G Rosenman, et al. Cardiac toxicity after radiotherapy for stage III non–small-cell lung cancer: Pooled analysis of dose-escalation trials delivering 70 to 90 Gy. *Journal of Clinical Oncology*, 35(13):1387, 2017.

John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10):1113, 2013.

Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. Representation learning for treatment effect estimation from observational data. In *Advances in Neural Information Processing Systems*, pages 2633–2643, 2018.

Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GANITE: Estimation of individualized treatment effects using generative adversarial nets. *International Conference on Learning Representations (ICLR)*, 2018.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, pages 3391–3401, 2017.

Appendix A. Notation

In the table below, we summarise the notation used in our paper. Note that realisations of random variables are denoted using lowercase and subscripts/superscripts used with vector-valued functions denotes their output at the position of the given subscript/superscript.

$\mathcal{X}$	Feature space
$\mathcal{Y}$	Outcome space
$\mathcal{T}$	Intervention space
$\mathcal{W} = \{w_1, ..., w_k\}$	Set of treatments
$\mathcal{D}_w$	Dosage space for treatment $w \in \mathcal{W}$
$\mathbf{X} \in \mathcal{X}$	Features (random variable)
$Y : \mathcal{T} \rightarrow \mathcal{Y}$	Potential outcome function (function-valued random variable)
$T_f = (W_f, D_f) \in \mathcal{T}$	Factual/observed intervention (treatment-dosage pair) (random variable)
$Y_f \in \mathcal{Y}$	Outcome corresponding to the observed intervention ($Y_f = Y(W_f, D_f)$)
$\mathbf{G}$	Generator
$\mathbf{Z}$	Random noise (for input to generator) (random variable)
$\hat{Y}_{cf} : \mathcal{T} \rightarrow \mathcal{Y}$	Counterfactual outcome function induced by $\mathbf{G}$
$\mathbf{D}$	Discriminator
$\tilde{\mathcal{D}}_w = \{D^w_1, ..., D^w_{n_w}\}$	Random (finite) subset of $\mathcal{D}_w$
$n_w = \lvert\tilde{\mathcal{D}}_w\rvert$	Number of dosage levels passed to discriminator for treatment $w \in \mathcal{W}$)
$\tilde{\mathbf{Y}}_w = (D^w_i, \tilde{Y}^w_i)^{n_w}_{i=1}$	Vector of dosage-outcome pairs generated by $\mathbf{G}$ (and Y_f) using $\tilde{\mathcal{D}}_w$
$\mathbf{D}_\mathcal{W}$	Treatment discriminator
$\mathbf{D}_w$	Dosage discriminator for treatment $w \in \mathcal{W}$
$\mathbf{D}_H$	Hierarchical discriminator defined by combining $\mathbf{D}_\mathcal{W}$ and $\mathbf{D}_w$
$\mathbf{I}$	Inference network
$\mathcal{L}$	GAN loss
$\mathcal{L}_S$	Supervised loss

Appendix B. Counterfactual Generator Pseudo-code

Algorithm 1 Training of the generator in SCIGAN

```
**Input:** dataset $\mathcal{C} = \{(\mathbf{x}^i, t^i_f, y^i_f) : i = 1, ..., N\}$, batch size $n_{mb}$, number of dosages per treatment $n_d$, number of discriminator updates per iteration $n_D$, number of generator updates per iteration $n_G$, dimensionality of noise $n_z$, learning rate $\alpha$
**Initialize:** $\theta_G$, $\theta_{\mathcal{W}}$, $\{\theta_w\}_{w \in \mathcal{W}}$
**while G** has not converged **do**
  Discriminator updates
  **for** $i = 1, ..., n_D$ **do**
    Sample $(\mathbf{x}_1, (w_1, d_1), y_1), ..., (\mathbf{x}_{n_{mb}}, (w_{n_{mb}}, d_{n_{mb}}), y_{n_{mb}})$ from $\mathcal{C}$
    Sample generator noise $\mathbf{z}_j = (z^j_1, ..., z^j_{n_z})$ from Unif$([0,1]^{n_z})$ for $j = 1, ..., n_{mb}$
    **for** $w \in \mathcal{W}$ **do**
      **for** $j = 1, ..., n_{mb}$ **do**
        Sample $\tilde{D}^j_w = (d^{w,j}_1, ..., d^{w,j}_{n_d})$ independently and uniformly from $(\mathcal{D}_w)^{n_d}$
        Set $\tilde{\mathbf{y}}^j_w$ according to Eq. 5
      **end for**
      Calculate gradient of dosage discriminator loss
        $$g_w \leftarrow \nabla_{\theta_w} - \left[ \sum_{\{j : w_j = w\}} \sum_{k=1}^{n_d} \mathbb{I}_{\{d_j = d^{w,j}_k\}} \log \mathbf{D}_w(\mathbf{x}_j, \tilde{\mathbf{y}}^j_w) + \mathbb{I}_{\{d_j \neq d^{w,j}_k\}} \log(1 - \mathbf{D}_w(\mathbf{x}_j, \tilde{\mathbf{y}}^j_w)) \right]$$
      Update dosage discriminator parameters $\theta_w \leftarrow \theta_w + \alpha g_w$
    **end for**
    Set $\tilde{\mathbf{y}}_j = (\tilde{\mathbf{y}}^j_w)_{w \in \mathcal{W}}$
    Calculate gradient of treatment discriminator loss
      $$g_{\mathcal{W}} \leftarrow \nabla_{\theta_{\mathcal{W}}} - \left[ \sum_{j=1}^{n_{mb}} \sum_{w \in \mathcal{W}} \mathbb{I}_{\{w_j = w\}} \log \mathbf{D}_{\mathcal{W}}(\mathbf{x}_j, \tilde{\mathbf{y}}_j) + \mathbb{I}_{\{w_j \neq w\}} \log(1 - \mathbf{D}_{\mathcal{W}}(\mathbf{x}_j, \tilde{\mathbf{y}}_j)) \right]$$
    Update treatment discriminator parameters $\theta_{\mathcal{W}} \leftarrow \theta_{\mathcal{W}} + \alpha g_{\mathcal{W}}$
  **end for**
  Generator updates
  **for** $i = 1, ..., n_G$ **do**
    Sample $(\mathbf{x}_1, (w_1, d_1), y_1), ..., (\mathbf{x}_{n_{mb}}, (w_{n_{mb}}, d_{n_{mb}}), y_{n_{mb}})$ from $\mathcal{C}$
    Sample generator noise $\mathbf{z}_j = (z^j_1, ..., z^j_{n_z})$ from Unif$([0,1]^{n_z})$ for $j = 1, ..., n_{mb}$
    Sample $(\tilde{D}^j_w)_{w \in \mathcal{W}}$ from $\Pi_{w \in \mathcal{W}} (\mathcal{D}_w)^{n_d}$ for $j = 1, ..., n_{mb}$
    Set $\tilde{\mathbf{y}}$ according to Eq. 5
    Calculate gradient of generator loss
      $$g_G \leftarrow \nabla_{\theta_G} \left[ \sum_{j=1}^{n_{mb}} \sum_{w \in \mathcal{W}} \sum_{l=1}^{n_d} \mathbb{I}_{\{w_j = w, d_j = d^{w,j}_l\}} \log(\mathbf{D}^w_{\mathcal{W}}(\mathbf{x}_j, \tilde{\mathbf{y}}_j)_w \times \mathbf{D}^l_w(\mathbf{x}_j, \tilde{\mathbf{y}}^j_w)_l) \right.$$
      $$\left. + \mathbb{I}_{\{w_j \neq w, d_j \neq d^{w,j}_l\}} \log(1 - (\mathbf{D}^w_{\mathcal{W}}(\mathbf{x}_j, \tilde{\mathbf{y}}_j) \times D^l_w(\mathbf{x}_j, \tilde{\mathbf{y}}^j_w))) \right]$$
    Update generator parameters $\theta_G \leftarrow \theta_G + \alpha g_G$
  **end for**
**end while**
**Output: G**
```

Appendix C. Inference Network Pseudo-code

Algorithm 2 Training of the inference network in SCIGAN

```
**Input:** dataset $\mathcal{C} = \{(\mathbf{x}^i, t_f^i, y_f^i) : i = 1, ..., N\}$, trained generator $\mathbf{G}$, batch size $n_{mb}$, number of dosages per treatment $n_d$, dimensionality of noise $n_z$, learning rate $\alpha$
**Initialize:** $\theta_I$
**while I** has not converged **do**
    Sample $(\mathbf{x}_1, (w_1, d_1), y_1), ..., (\mathbf{x}_{n_{mb}}, (w_{n_{mb}}, d_{n_{mb}}), y_{n_{mb}})$ from $\mathcal{C}$
    Sample generator noise $\mathbf{z}_j = (z_1^j, ..., z_{n_z}^j)$ from Unif$([0, 1]^{n_z})$ for $j = 1, ..., n_{mb}$
    **for** $j = 1, ..., n_{mb}$ **do**
        **for** $w \in \mathcal{W}$ **do**
            Sample $\tilde{D}_w^j = (d_1^{w,j}, ..., d_{n_d}^{w,j})$ independently and uniformly from $(\mathcal{D}_w)^{n_d}$
            Set $\tilde{\mathbf{y}}_w^j$ according to Eq. 5
        **end for**
    **end for**
    Calculate gradient of inference network loss
        $$g_I \leftarrow -\nabla_{\theta_I} \left[ \sum_{j=1}^{n_{mb}} \sum_{w \in \mathcal{W}} \sum_{l=1}^{n_d} (\tilde{\mathbf{y}}_w^j)_l - \mathbf{I}(\mathbf{x}_j, (w, d_l^{w,j}))^2 \right]$$
    Update inference network parameters $\theta_I \leftarrow \theta_I + \alpha g_I$
**end while**
**Output: I**
```

Appendix D. Single Discriminator Model

In the paper we developed a hierarchical discriminator and demonstrated that it performs significantly better than the single discriminator setup that we now describe in this section.

D.1 Single Discriminator

In the single model, we will aim to learn a single discriminator, $\mathbf{D}$, that outputs $\mathbb{P}((W_f, D_f) = (w, d) | \mathbf{X}, \tilde{\mathcal{D}}_w, \tilde{\mathbf{Y}})$ for each $w \in \mathcal{W}$ and $d \in \tilde{\mathcal{D}}_w$. We will write $\mathbf{D}^{w,d}(\cdot)$ to denote the output of $\mathbf{D}$ that corresponds to the treatment-dosage pair (w, d). We define the loss, $\mathcal{L}_D$, to be

$$\mathcal{L}_D(\mathbf{D}; \mathbf{G}) = -\mathbb{E}\left[\sum_{w \in \mathcal{W}} \sum_{d \in \tilde{\mathcal{D}}_w} \mathbb{I}_{\{T_f = (w,d)\}} \log \mathbf{D}^{w,d}(\mathbf{X}, \tilde{\mathbf{Y}}) + \mathbb{I}_{\{T_f \neq (w,d)\}} \log(1 - \mathbf{D}^{w,d}(\mathbf{X}, \tilde{\mathbf{Y}}))\right] \quad (35)$$

where the expectation is taken over $\mathbf{X}$, $\{\tilde{\mathcal{D}}_w\}_{w \in \mathcal{W}}$, $\tilde{\mathbf{Y}}$, W_f and D_f and we note that the dependence on $\mathbf{G}$ is through $\tilde{\mathbf{Y}}$. Our single discriminator will be trained to minimise this loss directly. The generator GAN-loss, $\mathcal{L}_G$, is then defined by

$$\mathcal{L}_G(\mathbf{G}) = -\mathcal{L}_D(\mathbf{D}^*; \mathbf{G}) \quad (36)$$

where $\mathbf{D}^*$ is the optimal discriminator given by minimising $\mathcal{L}_D$. The generator will be trained to minimise $\mathcal{L}_G + \lambda \mathcal{L}_S$.

D.2 Single Discriminator Architecture

In the case of the single discriminator, we want the output of $\mathbf{D}$ corresponding to each treatment $w \in \mathcal{W}$, i.e. $(\mathbf{D}^{w,1}, ..., \mathbf{D}^{w,n_w})$, to be permutation equivariant with respect to $\tilde{\mathbf{y}}_w$ and permutation invariant with respect to each $\tilde{\mathbf{y}}_v$ for $v \in \mathcal{W} \setminus \{w\}$. To achieve this, we first define a function $f : \prod_{w \in \mathcal{W}} (\mathcal{D}_w \times \mathcal{Y})^{n_w} \rightarrow \mathcal{H}_S$ and require that this function be permutation invariant with respect to each of the spaces $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$. For each treatment, $w \in \mathcal{W}$, we introduce a multitask head, $f_w : \mathcal{X} \times \mathcal{H}_S \times (\mathcal{D}_w \times \mathcal{Y})^{n_w} \rightarrow [0, 1]^{n_w}$, and require that each of these functions be permutation equivariant with respect to their corresponding input space $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$ but they can depend on the features, $\mathbf{x} \in \mathcal{X}$, and invariant latent representation coming from f arbitrarily. Writing f_w^j to denote the jth output of f_w, the output of the discriminator given input features, $\mathbf{x}$, and generated outcomes, $\tilde{\mathbf{y}}$, is defined by

$$\mathbf{D}^{w,j}(\mathbf{x}, \tilde{\mathbf{y}}) = f_w^i(\mathbf{x}, f(\tilde{\mathbf{y}}), \tilde{\mathbf{y}}_w). \quad (37)$$

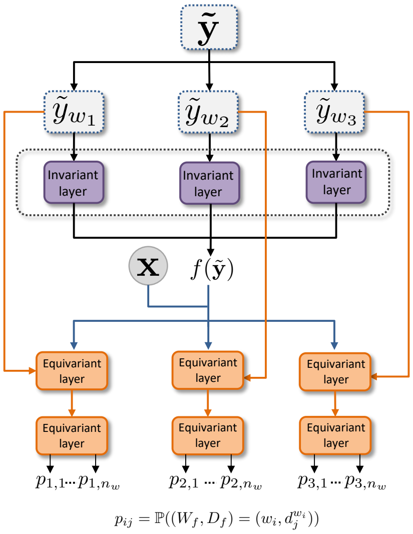


Figure 6: Overview of the single discriminator architecture.

To construct the function f, we concatenate the outputs of several invariant layers of the form given in Eq. (27) that each individually act on the spaces $(\mathcal{D}_w \times \mathcal{Y})^{n_w}$. That is, for each treatment, $w \in \mathcal{W}$ we define a map $f_{inv}^w : (\mathcal{D}_w \times \mathcal{Y})^{n_w} \rightarrow \mathcal{H}_S^w$ by substituting $\tilde{\mathbf{y}}_w$ for $\mathbf{u}$ in Eq. (27). We then define $\mathcal{H}_S = \prod_{w \in \mathcal{W}} \mathcal{H}_S^w$ and $f(\tilde{\mathbf{y}}) = (f_{inv}^{w_1}(\tilde{\mathbf{y}}_{w_1}), ..., f_{inv}^{w_k}(\tilde{\mathbf{y}}_{w_k}))$.

Each f_w will consist of two layers of the form given in Eq. (28) with the equivariance input, $\mathbf{u}$, to first layer being $\tilde{\mathbf{y}}_w$ and to the second layer being the output of the first layer and the auxiliary input, $\mathbf{v}$, to the first layer being the concatenation of the features and invariant representation, i.e. $(\mathbf{x}, f(\tilde{\mathbf{y}}))$ and then no auxiliary input to the second layer.

A diagram depicting the architecture of the single discriminator model can be found in Fig. 6.

Appendix E. Dataset descriptions

TCGA: The TCGA dataset consists of gene expression measurements for cancer patients (Weinstein et al., 2013). There are 9659 samples for which we used the measurements from the 4000 most variable genes. The gene expression data was log-normalized and each feature was scaled in the $[0, 1]$ interval. Moreover, for each patient, the features $\mathbf{x}$ were scaled to have norm 1. We give meaning to our treatments and dosages by considering the treatment as being chemotherapy/radiotherapy/immunotherapy and their corresponding dosages. The outcome can be thought of as the risk of cancer recurrence (Schwab et al., 2019).

News: The News dataset consists of word counts for news items. We extracted 10000 samples each with 2858 features. As in (Johansson et al., 2016; Schwab et al., 2019), we give meaning to our treatments and dosages by considering the treatment as being the viewing device (e.g. phone, tablet etc.) used to read the article and the dosage as being the amount of time spent reading it. The outcome can be thought of as user satisfaction.

MIMIC III: The Medical Information Mart for Intensive Care (MIMIC III) (Johnson et al., 2016) database consists of observational data from patients in the ICU. We extracted 3000 patients that receive antibiotics treatment and we used as features 9 clinical covariates measured during the day of ICU admission. Again, the features were scaled in the $[0, 1]$ interval. In this setting, we can considered as treatments the different antibiotics and their corresponding dosages.

For a summary description of the datasets, see table 6. The datasets are split into 64/16/20% for training, validation and testing respectively. The validation dataset is used for hyperparameter optimization.

	TCGA	**News**	**MIMIC**
Number of samples	9659	10000	3000
Number of features	4000	2858	9
Number of treatments	3*	3	2

Table 6: Summary description of datasets. *: for our final experiment in Appendix 7.6 we increase the number of treatments in TCGA to 6 and 9.

Appendix F. Benchmarks

We use the publicly available GitHub implementation of DRNet provided by Schwab et al. (2019): `https://github.com/d909b/drnet`. Moreover, we also used a GPS implementation similar to the one from `https://github.com/d909b/drnet` which uses the `causaldrf` R package (Galagate, 2016). More spcifically, the GPS implementation uses a normal treatment model, a linear treatment formula and a 2-nd degree polynomial for the outcome. Moreover, for the TCGA and News datasets, we performed PCA and only used the 50 principal components as input to the GPS model to reduce computational complexity.

Hyperparameter optimization: The validation split of the dataset is used for hyperparameter optimization. For the DRNet benchmarks we use the same hyperparameter optimization proposed by Schwab et al. (2019) with the hyperparameter search ranges described in Table 7. For SCIGAN, we use the hyperparameter optimization method proposed in GANITE (Yoon et al., 2018), where we use the complete dataset from the counterfactual generator to evaluate the MISE on the inference network. We perform a random search (Bergstra and Bengio, 2012) for hyperparameter optimization over the search ranges in Table 8. For a fair comparison, for the MLP-M model we used the same architecture used in the inference network of SCIGAN. Similarly, for the MLP model we use the same architecture as for the MLP-M, but without the multitask heads.

Hyperparameter	Search range
Batch size	32, 64, 128
Number of units per hidden layer	24, 48, 96, 192
Number of hidden layers	2, 3
Dropout percentage	0.0, 0.2
Imbalance penalty weight*	0.1, 1.0, 10.0
	Fixed
Number of dosage strata E	5

Table 7: Hyperparameters search range for DRNet. *: For the DRNet model using Wasserstein regularization only.

Hyperparameter	Search range
Batch size	64, 128, 256
Number of units per hidden layer	32, 64, 128
Size of invariant and equivariant representations	16, 32, 64, 128
	Fixed
Number of hidden layers per multitask head	2
Number of dosage samples	5
λ	1
Optimization	Adam Moment Optimization

Table 8: Hyperparameters search range for SCIGAN.

The hyperparameters used to generate the results for SCIGAN are given in Table 9.

The experiments were run on a system with 6CPUs, an Nvidia K80 Tesla GPU and 56GB of RAM.

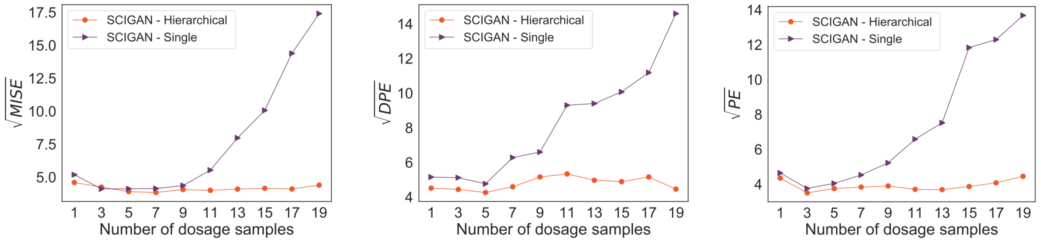
Hyperparameter	TCGA	News	MIMIC
Batch size	128	256	128
Number of units per hidden layer	64	128	32
Size of invariant and equivariant representations	16	32	16
Number of hidden layers per multitask head	2	2	2
Number of dosage samples	5	5	5
λ	1	1	1

Table 9: Hyperparameters used for obtaining results.

Appendix G. Investigating hyperparameter sensitivity (n_w)

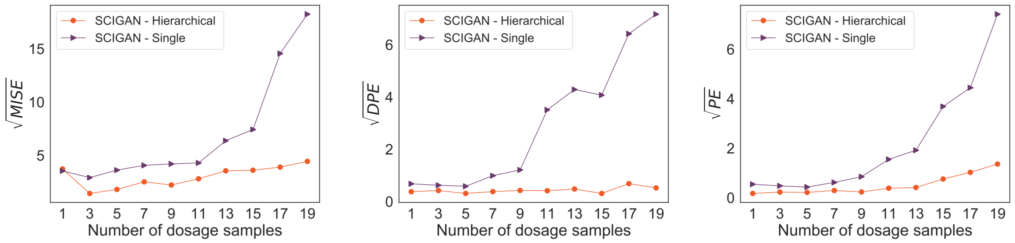
The performance of the single discriminator causes significant performance drops around $n_w = 9$ across all metrics. As previously noted, this is due to the dimension of the output space (which for $n_w = 9$ is 27) being too large. Conversely, we see that our hierarchical discriminator shows much more stable performance even when $n_w = 19$.

Here we present additional results for our investigation of the hyperparameters n_w. Fig. 7 reports each of the 3 performance metrics as we increase the number of dosage samples, n_w, used to train the discriminators on the News dataset. As with the TCGA results in the main paper we see that the single discriminator suffers a significant performance decrease when n_w is set too high.



(a) $\sqrt{\text{MISE}}$ (b) $\sqrt{\text{DPE}}$ (c) $\sqrt{\text{PE}}$

Figure 7: Performance of single vs. hierarchical discriminator when increasing the number of dosage samples (n_w) on News dataset.



(a) $\sqrt{\text{MISE}}$ (b) $\sqrt{\text{DPE}}$ (c) $\sqrt{\text{PE}}$

Figure 8: Performance of single vs. hierarchical discriminator when increasing the number of dosage samples (n_w) on TCGA dataset.