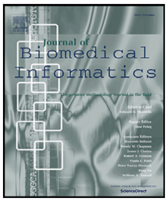
Contents lists available at ScienceDirect

Journal of Biomedical Informatics

journal homepage: www.elsevier.com/locate/yjbin



Original Research

Learning end-to-end patient representations through self-supervised covariate balancing for causal treatment effect estimation

Gino Tesei, Stefanos Giampanis, Jingpu Shi, Beau Norgeot [*]

Elevance Health, Palo Alto, CA 94301, USA

ARTICLE INFO

Keywords:
Causal treatment effect estimation
Representation learning
Self-supervised learning
Deep learning
Randomized controlled trials
Observational data

ABSTRACT

A causal effect can be defined as a comparison of outcomes that result from two or more alternative actions, with only one of the action-outcome pairs actually being observed. In healthcare, the gold standard for causal effect measurements is randomized controlled trials (RCTs), in which a target population is explicitly defined and each study sample is randomly assigned to either the treatment or control cohorts. The great potential to derive actionable insights from causal relationships has led to a growing body of machine-learning research applying causal effect estimators to observational data in the fields of healthcare, education, and economics. The primary difference between causal effect studies utilizing observational data and RCTs is that for observational data, the study occurs after the treatment, and therefore we do not have control over the treatment assignment mechanism. This can lead to massive differences in covariate distributions between control and treatment samples, making a comparison of causal effects confounded and unreliable. Classical approaches have sought to solve this problem piecemeal, first by predicting treatment assignment and then treatment effect separately. Recent work extended part of these approaches to a new family of representation-learning algorithms, showing that the upper bound of the expected treatment effect estimation error is determined by two factors: the outcome generalization-error of the representation and the distance between the treated and control distributions induced by the representation. To achieve minimal dissimilarity in learning such distributions, in this work we propose a specific auto-balancing, self-supervised objective. Experiments on real and benchmark datasets revealed that our approach consistently produced less biased estimates than previously published state-of-the-art methods. We demonstrate that the reduction in error can be directly attributed to the ability to learn representations that explicitly reduce such dissimilarity; further, in case of violations of the positivity assumption (frequent in observational data), we show our approach performs significantly better than the previous state of the art. Thus, by learning representations that induce similar distributions of the treated and control cohorts, we present evidence to support the error bound dissimilarity hypothesis as well as providing a new state-of-the-art model for causal effect estimation.

1. Introduction

Causal effect estimation of a binary exposure on a continuous outcome from observational data is a fundamental problem faced by many researchers in a broad range of diverse disciplines. For example, in social economy, policy makers need to determine who would benefit most from subsidized job training. In precision medicine, doctors need to decide which medication will cause better outcomes for a specific patient affected by a disease, taking into account relevant information such as age and pre-existing chronic conditions. The gold standard for estimating causal relationships has been randomized controlled trials (RCTs), which have three distinct stages: selection criteria to ensure that all samples are equivalent prior the study starting so that differences in outcomes can be attributed to differences in treatments, randomization of each sample to a treatment arm, and outcome comparison between the treatment arms. In particular, by controlling the

Abbreviations: AIPW, Augmented Inverse Propensity Weighted; ATE, Average Treatment Effect; BCAUSS, **B**alancing **C**ovariates **A**utomatically **U**sing **S**elf-**S**upervision; BCE, Binary Cross Entropy; DR, Doubly robust; ELU, Exponential Linear Unit; EHR, Electronic Health Record; GAN, Generative Adversarial Network; IHDP, Infant Health and Development Program; IPM, Integral Probability Metric; IPTW, Inverse Probability Treatment Weighting; ITE, Individual Treatment Effect; KS, Kolmogorov–Smirnov; MMD, Maximum Mean Discrepancy; ReLU, Rectified Linear Unit; RCT, Randomized Controlled Trial; SGD, Stochastic Gradient Descent; Tanh, Hyperbolic Tangent; VAE, Variational Autoencoder

[*] Corresponding author.

E-mail addresses: gino.tesei@carelon.com (G. Tesei), stefanos.giampanis@carelon.com (S. Giampanis), jingpu.shi@carelon.com (J. Shi), beau.norgeot@carelon.com (B. Norgeot).

https://doi.org/10.1016/j.jbi.2023.104339
Received 19 December 2022; Received in revised form 15 February 2023; Accepted 13 March 2023
Available online 20 March 2023

treatment assignment mechanism, RCTs reduce the bias of treatment effect estimation due to factors that affect both treatment and outcome (*confounders*). However, RCTs are not always feasible due to logistical, ethical, or financial considerations. Moreover, being based on restricted populations following strict protocols that frequently do not match daily standards of care, the results from RCTs do not always generalize to new patients in the real world [1,2]. In the past decade, the viability of observational data to infer causal relationships has been explored due to the increasingly available patient data captured in electronic health records (EHRs), the remarkable advances of machine learning techniques, and considerably reduced cost of such studies.

Classical works in causal inference addressed the problem of estimating average treatment effect (ATE) from observational data with covariate adjustment, also known as back-door adjustment [3–5], or weighting methods [6], where an estimate of the probability of treatment, conditioned on covariates (*propensity score*), is used to reweight the units in the observational data to make the treated and control populations more comparable. Targeted Learning [7] adjusts the estimation of an initial statistical model in a step targeted toward making an optimal bias–variance trade-off of the causal effect. None of those approaches are built or trained end-to-end, meaning that the models in each approach are trained separately without sharing a common representation. For example, in Targeted Learning, the initial statistical model is trained separately from the one that adjusts its estimation. Shalit et al. [8] extended part of those approaches to a new family of representation-learning based algorithms [9], and demonstrated that the expected error in learning individual treatment effect (ITE) is upper bounded by the error of learning factual and counterfactual outcomes plus a term depending on the dissimilarity of the treated and untreated distributions induced by the learned representation.

In this work, we propose a self-supervised auto-balancing objective specifically designed to minimize the dissimilarity of the learned representations for treated and untreated cohorts. We work under the common simplifying assumption of *no-hidden confounding*, i.e., assuming that all the factors that affect both treatment and outcome are observed. We call this method BCAUSS (**B**alancing **C**ovariates **A**utomatically **U**sing **S**elf-**S**upervision). Utilizing two widely used datasets in the casual inference community, namely IHDP and Jobs (see Section 2.5), we found that BCAUSS produced less biased estimates than previously published state-of-the-art methods. In particular, we compared BCAUSS to Dragonnet [10], the current state of the art on IHDP. We show that BCAUSS produced less biased estimates than Dragonnet because of its ability to learn less dissimilar treated and untreated distributions, consistently to how they should be in RCTs. In particular, we show that the balancing effect of our self-supervised auto-balancing objective is crucial when the positivity assumption is violated, which is often the case in observational data.

1.1. Related work

Classical causal modeling typically involves the concept of *balancing score* [11]. Back-door adjustment methods [3–5] and weighting methods [6] adopt a special balancing score, i.e., the *propensity score*, to re-weight the units in observational data to make the treated and control populations more comparable. Targeted Learning [7] adjusts the estimation of an initial statistical model with a second model, making an optimal bias–variance trade-off of the causal effect. Treatment effect estimation has also been approached by designing splitting criteria specific to treatment effect in recursive partitioning [12–14], adopting ensemble algorithms and meta algorithms [15,16]. Other approaches like Propensity Dropout [17] and Perfect Matching [18] combine (pre-trained) propensity scores with neural networks. In this work, we do not use propensity scores but an auto-balancing self-supervised objective to balance covariates between the treated and control cohorts.

Shalit et al. [8] introduced a new family of representation-learning based algorithms [9], including CFR and TARNet. Shi et al. [10] introduced Dragonnet, extending TARNet adopting a binary cross-entropy objective for propensity score estimation, and a targeted regularization objective to achieve asymptotically optimal properties. BCAUSS adopts the same multi-task network architecture of Dragonnet, CFR and TARNet, but with a specific auto-balancing self-supervised objective, which we show is critical to achieve minimal dissimilarity in learning treated and untreated distributions.

Belthangady and Norgeot [19] adopted a single-task network architecture to estimate propensity scores with binary cross-entropy as well as to balance covariates. In our work, we adopted a multi-task network architecture to estimate factual and counterfactual outcomes while balancing covariates at the same time and show that binary cross-entropy leads to sub-optimal results.

Other works applied deep generative models to casual inference. For example, CEVEA [20], GANITE [21], and CMPGP [22] use Variational Autoencoder (VAEs) [23], Generative Adversarial Nets (GANs) [24], and multi-task Gaussian processes, respectively, to estimate treatment effects. Shi et al. [25] adopted GANs for high-fidelity privacy-conscious patient data generation, while Zhang et al. [26] proposed a variational inference approach to infer latent factors from the observed variables. Our work does not aim to learn the joint probability distribution of covariates and outcome, but the conditional probability distribution of the outcome given covariates.

2. Materials and methods

Let assume a population where we can measure a vector of d covariates X for each individual and where each one of them is subject to a dichotomous treatment T (1: treated, 0: untreated), producing an outcome Y. Let Y_0 be the outcome variable that would have been observed under the treatment value $t = 0$, and Y_1 the outcome variable that would have been observed under the treatment value $t = 1$. We denote with P_n an observable sample of size n of such population, i.e. $P_n = \left\{y^{(i)}, t_i, \mathbf{x}_i\right\}_{i=1}^{n}$, where t_i and $\mathbf{x}_i$ are realizations of T and X, while $y^{(i)}$ is a realization of Y_0, if $t_i = 0$, or a realization of Y_1, if $t_i = 1$. Also, we denote with I_n the subset of the input realizations only, i.e. $I_n = \left\{t_i, \mathbf{x}_i\right\}_{i=1}^{n}$. Moreover, we assume that the following three fundamental assumptions for treatment effect estimations [11] are satisfied.

Assumption 1. (SUTVA) *The Stable Unit Treatment Value Assumption requires that the potential outcomes for one individual are unaffected by the treatment of others.*

Assumption 2. (Ignorability) *The distribution of treatment is conditional independent of the potential outcomes, given covariates, i.e.,*

$$T \perp\!\!\!\perp \left(Y_0, Y_1\right) | X$$

Assumption 3. (Positivity) *Every individual has a non-zero probability to receive either treatment or control, given covariates, i.e.,*

$$(\forall i)\left(0 < P\left(T = 1 \,\middle|\, X = \mathbf{x}_i\right) < 1\right).$$

In observational datasets we might observe violations of the positivity assumption, as patients with certain covariate combinations are not observed to receive a treatment of interest, which may arise from contraindications to treatment or small sample size [27,28]. When the ignorability and the positivity assumption hold, the treatment assignment is considered to be *strongly ignorable* [11], implying that among people with the same covariates, we can think of treatment as being randomly assigned. In the related causal graph this means that any backdoor path from t to Y is blocked. Hence, the average treatment effect (ATE) $\psi \in \mathbb{R}$ is defined as:

$$\psi = \mathbb{E}\left[Y_1 - Y_0\right] = \mathbb{E}\left[Y_1 \,|do\,(t=1)\right] - \mathbb{E}\left[Y_0 \,|do\,(t=0)\right],$$

where $do(t=1)$ or $do(t=0)$ denotes a manipulation on t by removing all its incoming edges on the related causal graph and setting $t = 1$ or $t = 0$, so that the effect of interest is causal [29]. To estimate ψ, the concept of *balancing score* is crucial in the sense of Rosenbaum and Rubin [11], i.e., a function $b(X)$ of the observed covariates X such that the conditional distribution of X given $b(X)$ is the same for treated and control units; that is, $X \perp\!\!\!\perp T \,|b(X)$. Indeed, in Rosenbaum and Rubin [11] the following theorem is proved.

Theorem 1. *Suppose treatment assignment is strongly ignorable and $b(X)$ is a balancing score. Then the expected difference in observed responses to the two treatments at $b(X)$ is equal to the average treatment effect at $b(X)$, that is*

$$\psi = \mathbb{E}\left[\mathbb{E}\left[Y_t\,|b(X),T=1\right] - \mathbb{E}\left[Y_t\,|b(X),T=1\right]\right] = \mathbb{E}\left[Y_1 - Y_0\,|b(X)\right].$$

2.1. Learning representations for causal treatment effect estimation

Following Shalit et al. [8], we will employ the following results and notations.

Definition 1. Let $\Phi : \mathcal{X} \to \mathcal{R}$ be a bijective representation function, where $\mathcal{X}$ is the space of covariates and $\mathcal{R}$ is the representation space. Let Ψ be the inverse of Φ, i.e.,for all $\mathbf{x} \in \mathcal{X}$ we have $\Psi(\Phi(\mathbf{x})) = \mathbf{x}$. Let $h : \mathcal{R}\times\{0,1\} \to \mathcal{Y} \subseteq \mathbb{R}$ the hypothesis.

Definition 2. For a representation function $\Phi : \mathcal{X} \to \mathcal{R}$, and for a distribution p defined over $\mathcal{X}$, let p_Φ be the distribution induced by Φ over $\mathcal{R}$. Define $p_\Phi^{t=1}(\mathbf{r}) := p_\Phi(\mathbf{r}|T=1)$, $p_\Phi^{t=0}(\mathbf{r}) := p_\Phi(\mathbf{r}|T=0)$, to be the treatment and control distributions induced over R.

Definition 3. Let $L : \mathcal{Y}\times\mathcal{Y} \to \mathbb{R}_+$ a loss function so that the expected loss for the unit and treatment pair $(\mathbf{x},t)$ is $l_{h,\Phi}(\mathbf{x},t) = \int_{\mathcal{Y}} L\left(Y_t, h(\Phi(\mathbf{x}),t)\right) p\left(Y_t|\mathbf{x}\right) dY_t$, the expected factual and counterfactual losses are:

$$\epsilon_F(h,\Phi) = \int_{\mathcal{X}\times\{0,1\}} l_{h,\Phi}(\mathbf{x},t)\,p(\mathbf{x},t)\,d\mathbf{x}dt$$

$$\epsilon_{CF}(h,\Phi) = \int_{\mathcal{X}\times\{0,1\}} l_{h,\Phi}(\mathbf{x},t)\,p(\mathbf{x},1-t)\,d\mathbf{x}dt,$$

from which the expected factual treated and control losses are $\epsilon_F^{t=1}(h,\Phi) = \int_{\mathcal{X}} l_{h,\Phi}(\mathbf{x},t)\,p^{t=1}(\mathbf{x})\,d\mathbf{x}$ and $\epsilon_F^{t=0}(h,\Phi) = \int_{\mathcal{X}} l_{h,\Phi}(\mathbf{x},t)\,p^{t=0}(\mathbf{x})\,d\mathbf{x}$, where $p^{t=1} := p(\mathbf{x}\,|T=1\,)$ and $p^{t=0} := p(\mathbf{x}\,|T=0\,)$, respectively.

Definition 4. Given that the ITE for unit $\mathbf{x}$ is $\tau(\mathbf{x}) := \mathbb{E}\left[Y_1 - Y_0|X=\mathbf{x}\right]$ and the treatment effect estimate of the hypothesis Q for unit $\mathbf{x}$ is $\tau_Q(\mathbf{x}) := Q(\mathbf{x},1) - Q(\mathbf{x},0)$, the expected Precision in Estimation of Heterogeneous Effect (PEHE) loss is

$$\epsilon_{PEHE}(Q) := \int_{\mathcal{X}} \left[\tau_Q(\mathbf{x}) - \tau(\mathbf{x})\right]^2 p(\mathbf{x})\,d\mathbf{x}.$$

Also, the expected variance of Y_t with respect to a distribution $p(\mathbf{x},t)$ is

$$\sigma^2_{Y_t}(p(\mathbf{x},t)) = \int_{\mathcal{X}\times\mathcal{Y}} \left[Y_t - \mathbb{E}\left(Y_t\,|\mathbf{x}\right)\right]^2 p(\mathbf{x},t)\,dY_t d\mathbf{x},$$

from which we define $\sigma^2_{Y_t} = \min\left\{\sigma^2_{Y_t}(p(\mathbf{x},t)), \sigma^2_{Y_t}(p(\mathbf{x},1-t))\right\}$ and $\sigma^2_Y = \min\left\{\sigma^2_{Y_0}, \sigma^2_{Y_1}\right\}$.

For two probability density functions p,q defined over $S \subseteq \mathbb{R}^d$ and for a function family G of functions $g : S \to \mathbb{R}$, we have that

$$IPM_G(p,q) := \sup_{g\in G}\left|\int_S g(s)\left[p(s) - q(s)\right]ds\right|.$$

Such integral probability metrics (IPM) are symmetric, obey the triangle inequality, $IPM_G(p,p) = 0$ and, for rich enough function families G, we have that $IPM_G(p,q) = 0 \Rightarrow p = q$. Hence, IPM is a metric over the corresponding set of probabilities. Examples of function families are the 1-Lipschitz functions [30] and the unit-ball of functions in a universal reproducing Hilbert kernel space [31]. In Shalit et al. [8] the following theorem is proved.

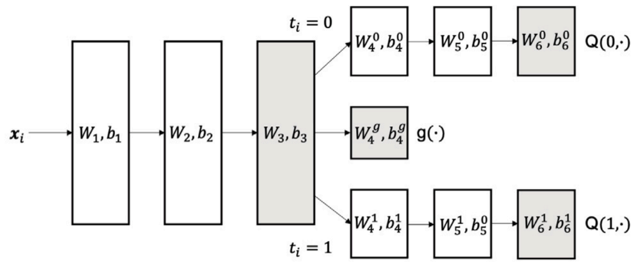


Fig. 1. BCAUSS and Dragonnet network architecture. The first three hidden layers are used to learn the internal representations, which is the input both for the balancing score estimator $g(\cdot)$ and the two separate control and treated sub-networks (three layers each). They both adopt mean squared error to train $Q(0,\cdot)$ and $Q(1,\cdot)$. However, while BCAUSS adopts an auto-balancing self-supervised objective to achieves balance of the learned representations, Dragonnet trains the propensity score estimator by using binary cross-entropy and a targeted regularization objective.

Theorem 2. *Under the above definitions and assuming the loss L is the squared loss, assuming there exists a constant $B_\Phi > 0$, such that for fixed $t \in \{0,1\}$, $\frac{1}{B_\Phi} l_{h,\Phi}(\Psi(\mathbf{r}),t) \in G$, we have*

$$\epsilon_{PEHE}(h,\Phi) \le 2\left(\epsilon_F^{t=0}(h,\Phi) + \epsilon_F^{t=1}(h,\Phi) + B_\Phi IPM_G\left(p_\Phi^{t=0}, p_\Phi^{t=1}\right) - 2\sigma_Y^2\right).$$

Theorem 2 states that the expected error in learning ITEs is upper bounded by the error of learning Y_0 and Y_1, plus the IPM term, which depends on the dissimilarity of the learned treated and untreated distributions induced by the representation. The minimal upper bound for a model with given factual treated and untreated losses is the one obtained when the IPM term is 0, reducing the causal treatment effect estimation problem to a standard regression problem.

2.2. BCAUSS

Fig. 1 depicts the three-task network architecture of BCAUSS, which is the same as Dragonnet [10]. For network layer $j \in \{1,2,3\}$ and sample $i \in \{1,2,\dots,n\}$, assuming $\mathbf{a}_{i,0} = \mathbf{x}_i$, we have $\mathbf{z}_{i,j} = \mathbf{W}_j\mathbf{a}_{i,j-1} + \mathbf{b}_j$ and $\mathbf{a}_{i,j} = ReLU\left(\mathbf{z}_{i,j}\right)$, where we use $ReLU(\cdot)$ to denote the ReLU activation function [32]. Further, $g_\theta\left(\mathbf{x}_i\right) = \sigma\left(\mathbf{W}_4^g\mathbf{a}_{i,3} + \mathbf{b}_4^g\right)$, where $\sigma(\cdot)$ is the sigmoid function and θ are all the network parameters. Notice that $g_\theta(\cdot)$ does not depend on T, which only enters the network after layer 3 for $Q(0,\cdot)$ and $Q(1,\cdot)$, but not for $g_\theta(\cdot)$. For network layer $j \in \{4,5\}$ and treatment $t_i \in \{0,1\}$, we have $\mathbf{z}_{i,j}^{t_i} = \mathbf{W}_{\mathbf{j}}^{t_i}\mathbf{a}_{i,j-1}^{t_i} + \mathbf{b}_{\mathbf{j}}^{t_i}$ and $\mathbf{a}_{i,j}^{t_i} = ReLU\left(\mathbf{z}_{i,j}^{t_i}\right)$. Hence, the treatment outcome estimate $Q_\theta\left(t_i,\mathbf{x}_i\right) = \mathbf{W}_{\mathbf{6}}^{t_i}\mathbf{a}_{i,5}^{t_i} + \mathbf{b_6}^{t_i}$. We adopt mean squared error for factual loss, i.e.,

$$L_{REG}^{(i)}\left(\theta; y^{(i)}, t_i, \mathbf{x}_i\right) = \left(1-t_i\right)\left(y^{(i)} - Q_\theta\left(0,\mathbf{x}_i\right)\right)^2 + \left(t_i\right)\left(y^{(i)} - Q_\theta\left(1,\mathbf{x}_i\right)\right)^2. \tag{1}$$

To train the balancing score estimator, instead of adopting the binary cross-entropy objective assuming as label t_i, as it happens in literature including Shi et al. [10], we adopt the following *auto-balancing self-supervised term*

$$L_{BAL}\left(\theta; I_n\right) = \frac{1}{d}\sum_{1\le k\le d}\left(\frac{\sum_{1\le i\le n}\frac{t_i}{g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\le i\le n}\frac{t_i}{g(\mathbf{x}_i)}} - \frac{\sum_{1\le i\le n}\frac{1-t_i}{1-g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\le i\le n}\frac{1-t_i}{1-g(\mathbf{x}_i)}}\right)^2, \tag{2}$$

where $x_{i,k}$ is kth covariate of individual i. Notice that such term does not come with a superscript, implying that to be computed the entire trainset is required. Specifically, we call such term *auto-balancing*, as we are imposing the constraint that the learned representation achieves balance. In practice this is obtained by minimizing the squared deviation between a reweighed treated and untreated cohort. Also, we use the term *self-supervised* as it depends only on the decision of treatment T and covariates X, which are input variables, and does not depend on potential outcomes Y_0, Y_1. Hence, the optimization problem can be formulated as:

$$\hat{\theta} = \arg\min_{\theta} J\left(\theta; P_n\right), \tag{3}$$

$$J\left(\theta; P_n\right) = \lambda_{BAL} L_{BAL}\left(\theta; I_n\right) + \frac{1}{n}\sum_{1\leq i\leq n} L^{(i)}_{REG}\left(\theta; y^{(i)}, t_i, \mathbf{x}_i\right), \tag{4}$$

where λ_{BAL} controls the relative importance of the two objective terms.

We show in Section and 2.4 and 4.1 that if the positivity assumption is satisfied then BCAUSS objective (Eq. (2)) incentivizes the network to recover a propensity score, since the propensity score minimizes our self-supervised objective ($L_{BAL} = 0$), leading to similar treated and untreated representations, and minimizing the IPM term of Theorem 2. However, positivity violations are frequent in observational data [27,28], and we show in Sections 2.4 and 4.1 that it is under these circumstances that BCAUSS performs significantly better than the previous state of the art.

2.3. Comparator: dragonnet

Dragonnet [10] is based on the same network architecture depicted in Fig. 1. However, while mean squared error is adopted for factual loss (Eq. (1)) to train the outcome estimator, the binary cross-entropy objective is adopted to train the propensity score estimator, i.e.,

$$L^{(i)}_{BCE}\left(\theta; t_i, \mathbf{x}_i\right) = t_i \log\left(g\left(\mathbf{x_i}\right)\right) + \left(1 - t_i\right)\log\left(1 - g\left(\mathbf{x_i}\right)\right). \tag{5}$$

Additionally, to achieve asymptotically robustness and efficiency, a further targeted regularization termis used, i.e.,

$$L^{(i)}_{T-REG}\left(\theta; y^{(i)}, t_i, \mathbf{x}_i\right) = \left[y^{(i)} - Q\left(t_i, \mathbf{x}_i\right) + \epsilon\left(\frac{t_i}{g\left(\mathbf{x_i}\right)} + \frac{1 - t_i}{1 - g\left(\mathbf{x_i}\right)}\right)\right]^2, \tag{6}$$

where ϵ is a hyperparameter. Hence, the optimization problem can be formulated as:

$$\hat{\theta} = \arg\min_{\theta} J'\left(\theta; P_n\right), \tag{7}$$

$$J'\left(\theta; P_n\right) = \frac{1}{n}\sum_{1\leq i\leq n} L^{(i)}_{REG} + \lambda_{BCE} L^{(i)}_{BCE} + \lambda_{TAR} L^{(i)}_{T-REG} \tag{8}$$

where $\lambda_{BCE}, \lambda_{TAR}$ control the relative importance of the three objective terms. Notice that not only BCAUSS and Dragonnet are based on the same architecture, but they have the same factual losses. Therefore, the upper bound of the expected error in learning ITEs of Theorem 2 will depend only on the IPM term, i.e., on the dissimilarity of treated and untreated representations. If the positivity assumption is satisfied, then Dragonnet also incentivizes the network to learn a function $g(X)$ that minimizes the IPM term. However, we show in Sections 2.4 and 4.1 that, when the positivity assumption is not satisfied (often the case for observational data), the binary cross-entropy objective (Eq. (5)) can be very confident in regions where the positivity does not hold, assuming values close to either one or zero, leading to different treated and untreated learned distributions. Further, for large values of λ_{BCE} Dragonnet is even less incentivized to recover a balancing score, since the relative importance of the binary cross-entropy objective is high. On the contrary, BCAUSS will still recover a balancing score when λ_{BAL} is large, recovering the values of X nearly exactly. This further implies that the accuracy of BCAUSS will be less sensitive to the choice of λ_{BAL} (see ablation experiments of Section 3.1), which is a desired property for a treatment effect estimation method, that should perform well across different datasets with minimum requirements for parameter tuning.

2.4. Why the positivity assumption plays so important role

If the positivity assumption is satisfied, the binary cross-entropy objective (Eq. (5)) and the BCAUSS objective (Eq. (2)) will tend to learn the same function $g(\cdot)$, i.e., our balancing score is the propensity score after the network is trained because the propensity score minimizes our self-supervised auto-balancing objective (see Rosenbaum [33]). However, when the positivity assumption is not satisfied, the binary cross-entropy objective and our self-supervised auto-balancing objective learn very different functions. The former seeks to maximize the accuracy of predicting whether a given patient is assigned to the treatment or control group and, in regions where the positivity does not hold, such an estimator can be very confident, assuming values close to either one or zero. Such values do not minimize our auto-balancing self-supervised objective, but lead to different treated and untreated learned distributions. It is under these circumstances that BCAUSS performs significantly better than the previous state of the art. Indeed, even if the positivity is violated, BCAUSS still seeks to achieve balance of the learned representations by weighting treated and untreated patients such that the difference between their reweighed averages is much closer.

To understand why this happens, let us consider the illustrative observational scenario in Fig. 2, having one covariate only (age) with the positivity assumption violated. We assume the network is constrained to learn function $g(\cdot)$ in the form of one of the two functions on the left side of Fig. 3. Their corresponding predicted propensity distributions are depicted on the right side of the figure, showing that the function in (B) produces much more similar treated and untreated distributions than (A) does. Which function would a binary cross-entropy estimator and BCAUSS learn? A binary cross-entropy based estimator would learn the function in (A) because it assigns low probability of being treated to individuals 40 or younger, and high probability to individuals older than 40, which is correct and close to the ground truth of how treatment is actually assigned to patients. Also, the function in (A) would minimize the binary cross-entropy objective (Eq. (5)). In contrast, the formulation of the loss function in Eq. (2) would enable BCAUSS to learn the function in (B) because this function weights the treated and control groups in such a way that the weighted mean of the two groups are pulled closer toward each other, thus minimizing the loss function in Eq. (2). It would not learn the function in (A) because this function gives the same weight to all the samples within each cohort and as a result, the loss function in Eq. (2) would be just the simple difference of the mean between the treated and control groups, implying a much larger balancing loss. Indeed, $L_{BAL} \approx 380$ if function (A) is learned, and $L_{BAL} \approx 54$ if function (B) is learned.

2.5. Benchmark and real-world datasets

2.5.1. Illustrative example

To gain an intuitive understanding of how our method works, we provide an illustrative example of estimating the causal effect of regular physical exercise (defined as ≥ 150 minutes/week) on people's health using observational data, controlling for possible confounding variables. As a marginal estimate, the ATE is relevant for public health policy, in that it quantifies the effect of regular physical exercise at the population level. In our simulated data, the 4 variables considered are denoted as $X = \{X_1, X_2, X_3, X_4\}$, where X_1(gender) is a Bernoulli variable with mean value 0.55, X_2(age) is a variable normally distributed with mean value 45 and standard deviation 25, X_3 (unweighted Charlson Comorbidity Index [34], or CCI) is a variable normally distributed with mean value 3.5 and standard deviation 2.5, and X_4 (smoking) is a Bernoulli variable with mean value $0.1 + 0.5\frac{X_2}{71} + 0.3\frac{X_3}{7}$, i.e., older persons or persons with higher comorbidity index are more likely to be smokers. We then consider two exposures: in the RCT scenario, A_{rct} is a binary indicator for regular physical exercise generated as a Bernoulli variable with mean value 0.50; in the observational scenario, A_{obs} is

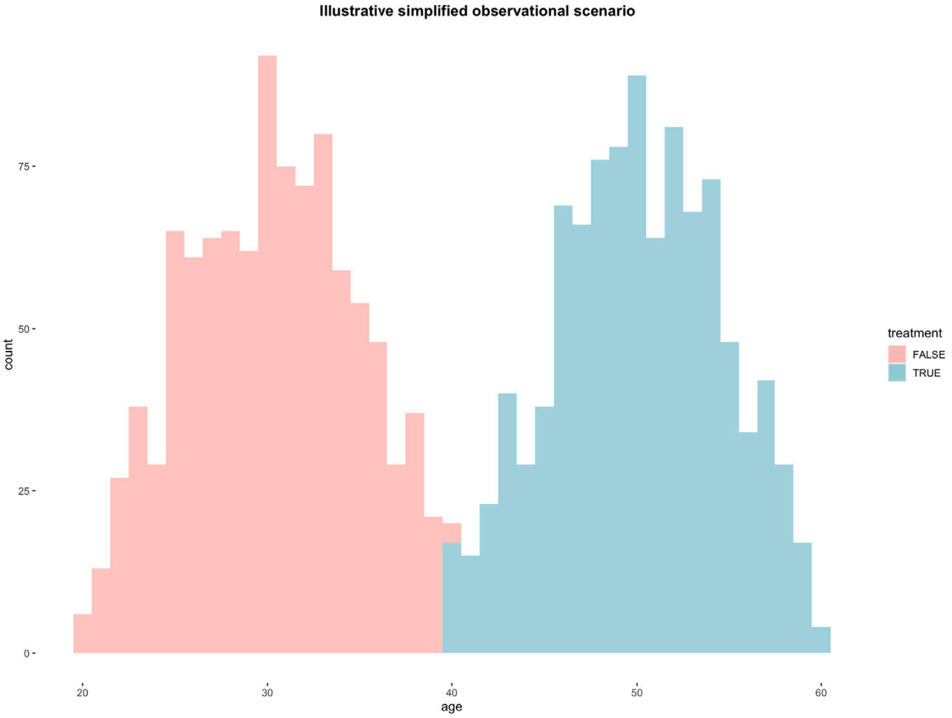


Fig. 2. Illustrative simplified observational scenario with one covariate only (age) with the positivity assumption violated. Younger individuals are more likely to be in the control group, where the average age is 30, while older patients are more likely to be in the treatment group, where the average age is 50.

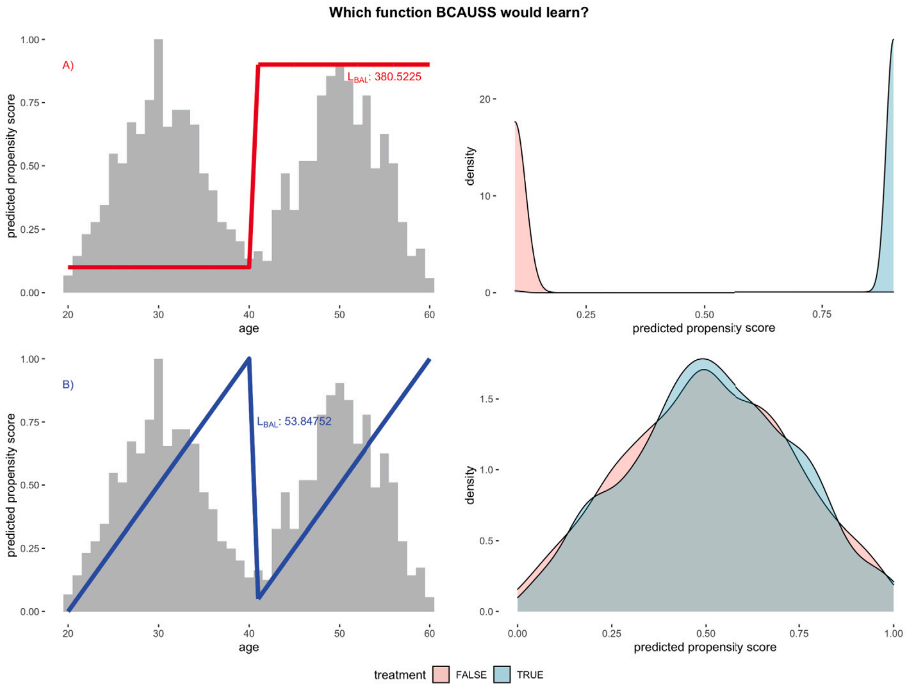


Fig. 3. Which function would a binary cross-entropy based estimator and BCAUSS learn? A binary cross-entropy based estimator would learn the function in (A) because it assigns low probability of being treated to individuals 40 or younger, and high probability to individuals older than 40, which is correct and close to the ground truth of how treatment is actually assigned to patients. However, this function implies treated and untreated representations highly dissimilar as shown on the right and, indeed, $L_{BAL} \approx 380$. In contrast, the formulation of the loss function in Eq. (2) would enable BCAUSS to learn the function in (B) because this function weighs the treated and control groups in such a way that the weighted mean of the two groups are pulled closer toward each other, thus minimizing the loss function in Eq. (2). It would not learn the function in (A) because this function gives the same weight to all the samples within each cohort and, as a result, the loss function in Eq. (2) would not be minimized. Indeed, if function (B) is learned, $L_{BAL} \approx 54$, and treated and untreated representations are more similar, as shown on the right.

also a binary exercise indicator, but generated as a Bernoulli variable with $logit\left(P\left(A_{obs}|X_2,X_3,X_4\right)\right) = -0.01 + 3.75\frac{X_2}{71} + 3.4\frac{X_3}{7} + 0.5X_4$ such that older persons, smokers and persons with higher comorbidity index are more likely to engage in regular physical exercise. The outcome (Y) is a binary variable with value one if the individual dies or is admitted to hospital within 3 years from the beginning of the study,

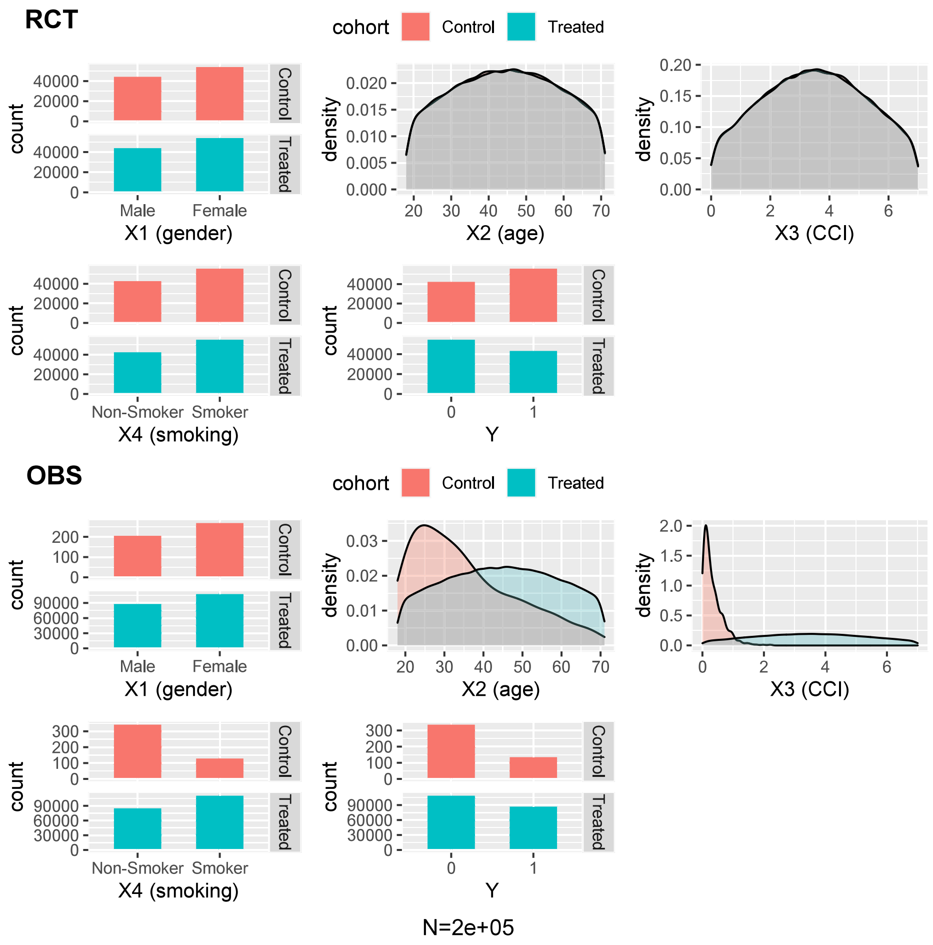


Fig. 4. Simulation to estimate the causal effect of regular physical exercise on hospitalization and mortality. **RCT**: random treatment assignment, where all covariates are identically distributed on both arms. **OBS**: observational scenario, where the younger, healthier, and non-smoking individuals are more prevalent in the control group while the older, diabetic, and smoking individuals are more prevalent in the treatment group. Note that for X_3 (Charlson Comorbidity Index), the positivity assumption is violated in the OBS scenario.

and zero otherwise. Age, comorbidity index, and smoking status are confounders, as they are associated with both the exposure and the outcome. The outcome is generated from a Bernoulli distribution with mean $0.09 - 0.11A - 0.03A(1 - X_4) + 0.25\frac{X_2}{71} + 0.3\frac{X_3}{7} + 0.3X_4$. With this outcome, regular physical exercise has the effect of reducing the risk of death or hospitalization by 0.14 among non-smokers and by 0.11 among smokers. Given a 56.47% rate of smokers in the sample, the true causal marginal effect of regular exercise in our simulated data is $0.56(-0.11) + 0.43(-0.14) = -0.12$.

In Fig. 4, we present the covariate and outcome distributions in the RCT scenario (top) and the observational scenario (bottom). Specifically, we can see that in the RCT scenario, all covariates are identically distributed on both arms. In the observational scenario, however, the younger, healthier, and non-smoking individuals are more likely in the control group while the older, diabetic, and smoking individuals are more likely in the treatment group. Also, we can see that the positivity assumption is violated for X_3. While the difference of the outcome between the treatment and control group in the RCT scenario provides correct estimation of the causal effect of regular exercise (−0.12), in the observational scenario, the risk of the treatment group is higher than that of the control group by 0.153, wrongly suggesting that engaging in regular physical exercise can lead to higher hospital admissions and death rate.

2.5.2. The IHDP benchmark dataset

The Infant Health and Development Program (IHDP) is a randomized controlled study designed to evaluate the effect of home visit from specialist doctors on the cognitive test scores of premature infants. The datasets is first used for benchmarking treatment effect estimation algorithms in Hill [35], where selection bias is induced by removing non-random subsets of the treated individuals to create an observational dataset, and the outcomes are generated using the original covariates and treatments. It contains 747 subjects and 25 variables. In order to compare our results with the literature and make our results reproducible, we used the simulated outcome implemented as setting “A” in Shalit et al. [8],Shi et al. [10], and downloaded the data at https://www.fredjo.com/, which is composed of 1000 repetitions of the experiment. We averaged our results over 1000 train/validation/test splits with ratios 70/20/10.

2.5.3. The jobs real-world dataset

The Jobs dataset by LaLonde [36] is a widely used benchmark in the causal inference community, where the treatment is job training and the outcomes are income and employment status after training. The dataset includes 8 covariates such as age, education, and previous earnings. Our goal is to predict unemployment, using the feature set of Dehejia and Wahba [37]. Following Shalit et al. [8], we combined the LaLonde experimental sample (297 treated, 425 control) with the PSID comparison group (2490 control). We averaged over 10 train/validation/test splits with ratio 62/18/20. The dataset is available for download at https://www.fredjo.com/.

2.6. Evaluation criteria

ATE. We adopt the ATE metric on IHDP to evaluate our model. As an established procedure [35,38–40], the ground truth of ATE can be calculated by averaging the differences of the outcomes in the treated and control groups. Then, comparing the ground truth ATE ($\tilde{\psi}_{ATE}$) with the related estimate obtained from a sample of the dataset, a method performance can be evaluated using the mean absolute error in ATE, i.e.,

$$\epsilon_{ATE} = \left| \tilde{\psi}_{ATE} - \frac{1}{n} \sum_{1 \leq i \leq n} Q\left(1, \mathbf{x}_i\right) - Q\left(0, \mathbf{x}_i\right) \right| .$$

ATT. We adopt the average treatment effect on the treated (ATT) metric on Jobs, because we can compute the true ATT on this particular dataset and use it to evaluate model performance. Since all the treated subjects T were part of the original randomized samples E, true ATT can be calculated following Shalit et al. [8] as: $\tilde{\psi}_{ATT} = \frac{1}{|T|} \sum_{i \in T} y^{(i)} - \frac{1}{|C \cap E|} \sum_{i \in C \cap E} y^{(i)}$, where C is the control group. Hence, $\epsilon_{ATT} = \left| \tilde{\psi}_{ATT} - \frac{1}{|T|} \sum_{i \in T} \left[Q\left(1, \mathbf{x}_i\right) - Q\left(0, \mathbf{x}_i\right) \right] \right|$.

IPMs. To measure the dissimilarity between treated and untreated distributions, we adopt Maximum Mean Discrepancy [31] and Wasserstein distance [41,42], which are the IPMs used in Shalit et al. [8] (see Theorem 2). Additionally, we calculate the Kolmogorov–Smirnov (KS) statistic [43,44] to show statistical significance of whether or not the treated and untreated samples have different distributions, i.e.,if p-value $> 1\%$, the difference between the two sample sets is not significant enough to say that they have different distributions.

2.7. Experimental details

In real-world scenarios, a good treatment effect estimation method should perform well across different datasets with minimum requirements for parameter tuning. Therefore, for BCAUSS we used exactly the same hyperparameters in IHDP and Jobs datasets, and with such hyperparameters our method achieved superior performance in both datasets. We showed separately in Section 3.1 (Ablation study) that these hyperparameters are optimal, and in Section 4.1 why they are optimal and should be adopted in general. With regard to the weights and bias in Fig. 1, we set $\mathbf{W}_i \in \mathbf{R}^{200 \times d}$ and $\mathbf{b}_i \in \mathbf{R}^{200 \times 1}$ for $i = 1$, $\mathbf{W}_i \in \mathbf{R}^{200 \times 200}$ and $\mathbf{b}_i \in \mathbf{R}^{200 \times 1}$ for $i \in \{2, 3\}$, $\mathbf{W}_i^g \in \mathbf{R}^{200 \times 1}$ and $\mathbf{b}_i^g \in \mathbf{R}^{1 \times 1}$ for $i = 4$, $\mathbf{W}_i^{t_i} \in \mathbf{R}^{200 \times 100}$ and $\mathbf{b}_i^{t_i} \in \mathbf{R}^{100 \times 1}$ for $i = 4$ and $t_i \in \{0, 1\}$, $\mathbf{W}_i^{t_i} \in \mathbf{R}^{100 \times 100}$ and $\mathbf{b}_i^{t_i} \in \mathbf{R}^{100 \times 1}$ for $i = 5$ and $t_i \in \{0, 1\}$, and finally $\mathbf{W}_i^{t_i} \in \mathbf{R}^{100 \times 1}$ and $\mathbf{b}_i^{t_i} \in \mathbf{R}^{1 \times 1}$ for $i = 6$ and $t_i \in \{0, 1\}$.

In the experiments, we adopted learning rate 1e-5, batch size equal to the train-set length, stochastic gradient descent with momentum [45] as optimization algorithm, and $\lambda_{BAL} = 1$. Models were trained on the optimal number of epochs by adopting early stopping [46].

3. Results

On IHDP, we compared BCAUSS with Dragonnet [10], the current state-of-the-art on this dataset to our knowledge. We also considered other methods adopted for comparison in Shi et al. [10]. Specifically, Balancing Neural Networks (BNN) [49] is a method that uses deep neural networks to learn treated and untreated representations balanced by minimizing their discrepancy; Counterfactual Regression (CFR Wass) [52] adopts deep neural networks to learn treated and untreated representations, regularizing them by minimizing their Wasserstein distance; Treatment-Agnostic Representation Network (TARNet) [8] is similar to CFR, but does not apply any regularization to its representations. Additionally, in our comparison we included the augmented inverse propensity weighted (AIPW) estimator [55,56] defined in Belthangady and Norgeot [19], which deploys a deep neural network as propensity score estimator and two linear regressors regularized with L2 penalty to learn outcomes for control and treatment respectively. Finally, we considered two deep generative methods: CEVAEs [48] adopts VAEs [23] to infer unobserved confounders such as socio-economic status, while GANITE [21] uses GANs [24] to infer ITEs.

On Jobs, we compared BCAUSS with GANITE [21], the current state-of-the-art on this dataset to our knowledge. We also considered other methods adopted for comparison in Yoon et al. [21] in addition to the ones already introduced for IHDP. Specifically, OLS/LR-1 [47] adopts ordinary least squares with treatment as a feature; OLS/LR-2 [47] uses separate regressors for each treatment; Balancing Linear Regression (BLR) [49] uses linear regression to obtain representations from features, minimizing their discrepancy to obtain balance; K-NN [50] estimates the counterfactual outcome by searching the nearest neighbors with most similar features but opposite treatments; BART [51] is a tree-based regression model adopting a Bayesian backfitting Markov Chain Monte Carlo (MCMC) algorithm for fitting and inference; Random Forest [53] is an ensemble learning method that constructs a large number of decorrelated weak learners and obtains the prediction as weighted average of their outputs; Causal Forest [54] is an extension of Random Forest [53] adopting asymptotic normality theory to obtain pointwise consistency.

We reproduced the experiments of Dragonnet, GANITE, CFR Wass, TARNet, AIPW, and quoted the performances of the remaining models from Shalit et al. [8].

The best performance on IHDP was achieved with BCAUSS model using only the self-supervised auto-balancing term (Eq. (2)) to train the balancing score estimator. When adding the Dragonnet targeted regularization term(Eq. (6)), BCAUSS produces sub-optimal results regardless of whether a binary cross-entropy loss is added or not. Using only the binary cross-entropy also produces sub-optimal results. In Appendix A we formally analyzed the effect of back-propagating the gradient with respect to BCAUSS's self-supervised, auto-balancing objective, and compared it to the binary cross-entropy objective in Appendix B. In Section 4.1 we demonstrate why the auto-balancing objective produces a lower upper bound of the expected error in learning ITE (Theorem 2) than binary cross-entropy. We also show that the error margin increases as the treated and untreated covariate distributions become highly dissimilar or the positivity assumption is violated, which is common in observational data.

On Jobs BCAUSS showed the best out-sample performance (Table 1), while the best in-sample performance was obtained by AIPW.

3.1. Ablation experiments to test hyperparameter importance

The models shown in Table 2 refer to the best BCAUSS models that we trained. These models adopt the experimental settings specified in Section 2.7 except the ones explicitly stated in each row of Table 2.

3.1.1. Effect of optimizer and batch size

The effect of the optimizer can be analyzed by comparing row 5–8 (SGD) to row 9–12 (Adam) in Table 2. The SGD optimizer consistently improves ATE estimation compared to Adam by 0.05 in-sample and 0.08 out-of-sample, except for the lowest batch size where the out-of-sample difference is 0.07. The effect of batch size can be analyzed by comparing row 5 to row 6–8 for SGD and row 9 to 10–12 for Adam in Table 2. While for Adam we do not observe any test performance improvement with larger batch sizes, for SGD we do observe an improvement of 0.01 for batch size equal or higher than $\left\lfloor \frac{5}{12} \cdot n \right\rfloor$. In Appendix A we discuss in detail why larger batch-size with SGD is beneficial.

Table 1
Results on IHDP (left) and Jobs (right). Left: BCAUSS is state-of-the-art on the IHDP benchmark dataset. “+BCE” and “+T_REG” mean adding the binary cross-entropy of Eq. (5) and the targeted regularization term of Eq. (6) to the overall objective. Similarly, “-T_REG” means removing the targeted regularization term of Eq. (6) from the objective. Right: BCAUSS showed the best out-sample performance, while the best in-sample performance was obtained by AIPW. Lower is better.

Method	IHDP		Method	Jobs	
	ϵ^{tr}_{ATE}	ϵ^{te}_{ATE}		ϵ^{tr}_{ATT}	ϵ^{te}_{ATT}
GANITE [21]	0.43±.05	0.49 ±.05	OLS/LR-1 [47]	.01±.00	.08±.04
CEVAEs [48]	0.34±.01	0.46±.02	OLS/LR-2 [47]	.01±.01	.08±.03
AIPW [19]	0.13±.00	0.29±.01	BLR [49]	.01±.01	.08±.03
BNN [49]	0.37±.03	0.42±.03	k-NN [50]	.21±.01	.13±.05
TARNet [8]	0.26±.01	0.28±.01	BART [51]	.02±.00	.08±.03
CFR Wass [52]	0.25±.01	0.27±.01	Rand. Forest [53]	.03±.01	.09±.04
Dragonnet [10]	0.14±.01	0.20±.01	Caus. Forest [54]	.03±.01	.09±.04
baseline (Dragonnet) -T_REG	0.13±.00	0.21±.01	AIPW [19]	.00±.01	.09±.02
baseline (Dragonnet)	0.15±.01	0.20±.01	BNN [49]	.04±.01	.09±.04
BCAUSS +T_REG	0.15±.00	0.19±.01	TARNet [8]	.05±.02	.11±.04
BCAUSS +BCE +T_REG	0.16±.00	0.20±.01	CFR Wass [52]	.04±.01	.09±.03
BCAUSS +BCE	0.13±.00	0.17±.01	GANITE [21]	.01±.01	.06±.03
BCAUSS	**0.10±.00**	**0.15±.01**	BCAUSS	.02±.00	**.05±.02**

Table 2
Ablation results of the different variants described in Section 3.1 on IHDP dataset (n is the total number of observations on train-set). $\lambda_{BAL} = 1$ corresponds to add to the objective the self-supervised auto-balancing term of Eq. (2), $\lambda_{BCE} = 1$ corresponds to add to the objective the binary cross-entropy term of Eq. (5), $\lambda_{TAR} = 1$ corresponds to add to the objective the targeted regularization termof Eq. (6). Row 5 with ReLU activation function corresponds to BCAUSS. Row 0 with ELU activation function corresponds to Dragonnet. Bold indicates the best performance overall.

Model	Optimizer	Batch size	λ_{BCE}	λ_{BAL}	λ_{TAR}	ReLU		ELU		Tanh	
						ϵ^{tr}_{ATE}	ϵ^{te}_{ATE}	ϵ^{tr}_{ATE}	ϵ^{te}_{ATE}	ϵ^{tr}_{ATE}	ϵ^{te}_{ATE}
(0)	SGD	n	1	0	1	0.16±.01	0.20±.01	0.14±.01	0.18±.01	0.15±.01	0.19±.01
(1)	SGD	n	1	0	0	0.12±.00	0.16±.01	0.13±.00	0.17±.01	0.13±.00	0.18±.01
(2)	SGD	n	1	0.5	0	0.13±.00	0.16±.01	0.13±.00	0.17±.01	0.13±.00	0.18±.01
(3)	SGD	n	1	1	0	0.13±.00	0.16±.01	0.13±.01	0.17±.01	0.13±.00	0.18±.01
(4)	SGD	n	1	1.5	0	0.13±.00	0.17±.01	0.13±.01	0.17±.01	0.13±.00	0.18±.01
(5)	SGD	n	0	1	0	**0.10±.00**	**0.15±.01**	0.12±.00	0.16±.01	0.13±.00	0.17±.01
(6)	SGD	$\lfloor \frac{n}{2} \rfloor$	0	1	0	**0.10±.00**	**0.15±.01**	0.12±.00	0.16±.01	0.12±.00	0.17±.01
(7)	SGD	$\lfloor \frac{5}{12} \cdot n \rfloor$	0	1	0	**0.10±.00**	**0.15±.01**	0.12±.00	0.16±.01	0.12±.00	0.17±.01
(8)	SGD	$\lfloor \frac{n}{3} \rfloor$	0	1	0	**0.10±.00**	0.16±.01	0.12±.00	0.16±.01	0.12±.00	0.17±.01
(9)	Adam	n	0	1	0	0.15±.00	0.23±.01	0.15±.00	0.19±.01	0.15±.00	0.20±.01
(10)	Adam	$\lfloor \frac{n}{2} \rfloor$	0	1	0	0.15±.00	0.23±.01	0.15±.00	0.19±.01	0.15±.00	0.20±.01
(11)	Adam	$\lfloor \frac{5}{12} \cdot n \rfloor$	0	1	0	0.15±.00	0.23±.01	0.15±.00	0.19±.01	0.15±.00	0.20±.01
(12)	Adam	$\lfloor \frac{n}{3} \rfloor$	0	1	0	0.15±.00	0.23±.01	0.16±.00	0.20±.01	0.15±.00	0.20±.01
(13)	SGD	n	0	0.5	0	**0.10±.00**	**0.15±.01**	0.12±.00	0.16±.01	0.13±.00	0.17±.01
(14)	SGD	n	0	1.5	0	**0.10±.00**	**0.15±.01**	0.12±.00	0.16±.01	0.13±.00	0.17±.01

3.1.2. *Effect of activation function*

The effect of activation function can be analyzed by comparing the columns ReLU, ELU and Tanh in each row in Table 2, corresponding to in-sample and out-sample performance of the activation functions ReLUs, ELUs and Tanhs respectively. We find that overall the best performance is observed with ReLUs. Specifically, in case of SGD optimizer, ReLUs consistently outperform ELUs by 0.02 on training set and 0.01 on test-set. In turn, ELUs outperformTanhs by 0.01 on test-set. In case of Adam optimizer, while in-sample performance is pretty homogeneous across the three, in out-sample performance, ELUs outperform Tanhs by 0.01 on test-set. In turn, Tanhs outperform ReLUs by 0.03 on test-set.

3.1.3. *Effect of self-supervised auto-balancing term, binary cross-entropy term and targeted regularization term*

The effect of the self-supervised auto-balancing term can be analyzed by comparing row 0–4 to row 5–14 in Table 2. We can see that optimal train and test performance can be achieved when the network is regularized only with such term. Additionally, the same optimal results can be achieved with different values of λ_{BAL}. The effect of the binary cross-entropy term can be analyzed by comparing row 1–4 to row 5–12. For ReLUs, not adopting the binary cross-entropy improves the performance, at least, by 0.03 on trainset and 0.01 on test-set. The effect of the targeted regularization termcan be analyzed by comparing row 0 (Dragonnet) to row 1. Adopting such term is less optimal for all the activation functions. Note that our best test result for Dragonnet is better than the one reported in the original paper [10], obtained with a lower batch-size. In Appendix B we discuss in detail why larger batch-size when working with SGD is beneficial for the binary cross-entropy objective.

4. Discussion

Ideally in an RCT, patients are randomized in such a way that the distributions of covariates in the treated population match the covariate distributions in the control population. In other words, for a study with the covariates of age, sex, and height, ideal randomization should produce a treated population that has the same mean for age between the treated and untreated populations and likewise for height and percentage of each sex. This prospective balance is, of course, impossible to produce for retrospective, or observational, studies. However, prior work (Belthangady and Norgeot [19]) has demonstrated that while the observed covariates cannot be directly balanced, it is possible to use deep learning to learn weights for each observed covariate when accounting for treatment assignment, such that multiplying the observed covariate by their learned weight does generate balanced distributions between treatment and control and, furthermore, that doing so produces less biased ATE estimates when coupled with well-established propensity-based approaches [57]. However, this previous approach had two major limitations. First, it did not generate effect estimates itself but relied on downstream propensity-based estimators such as IPTW [33,58] or DR [59,60]. Second, it was incapable of producing ITEs, which are highly desirable for fields such as personalized medicine. BCAUSS extends the work of Belthangady and Norgeot [19] by addressing both limitations. This has been achieved by pairing the novel auto-balancing loss with the current state of the art deep learning architecture employed by Dragonnet for estimating ITEs. The resulting multi-task deep neural network has two objectives: minimizing factual loss, and minimizing the difference of the reweighted covariate means between the treated and untreated populations. While the network could have had a third objective, which would have been minimizing the binary cross-entropy loss for estimating each patient's treatment assignment, approach followed by both Belthangady and Norgeot [19] and Dragonnet, we show here that a binary cross-entropy objective is not only unnecessary, but it is actually deleterious.

Empirically, the results on IHDP and Jobs datasets shown in Section 3 revealed that BCAUSS consistently produced less biased estimates of causal treatment effect than the previously published state-of-the-art methods. Section 4.1 provides an understanding of how this method compares to methods adopting binary cross-entropy as propensity score estimator, using the dataset introduced in Section 2.5.1. In particular, we show why BCAUSS auto-balancing objective becomes truly interesting when treated and untreated covariate distributions are quite dissimilar and the positivity assumption is violated, which is often the case in real world data [27,28].

These findings connect very elegantly to Theorem 2, which states that the expected error in learning ITEs is bounded by the error of learning Y_0 and Y_1, plus the IPM term, which depends on the dissimilarity of the learned treated and untreated distributions induced by the representation. As BCAUSS and Dragonnet are not only based on the same architecture, but they have the same factual and counterfactual loss, to explain the superior performance of BCAUSS, we show in Section 4.2 that the learned treated and untreated distributions of BCAUSS are less dissimilar than the ones of Dragonnet on the IHDP dataset implying, therefore, a lower upper bound of the related expected error in learning ITEs. Further, in Section 4.3 the same analysis is repeated on the Jobs dataset, showing again that BCAUSS learns less dissimilar treated and untreated representations than Dragonnet. Finally, in Appendix A we formally analyze the effect of back-propagating the gradient with respect to our self-supervised auto-balancing objective, and compare it to the binary cross-entropy objective in Appendix B.

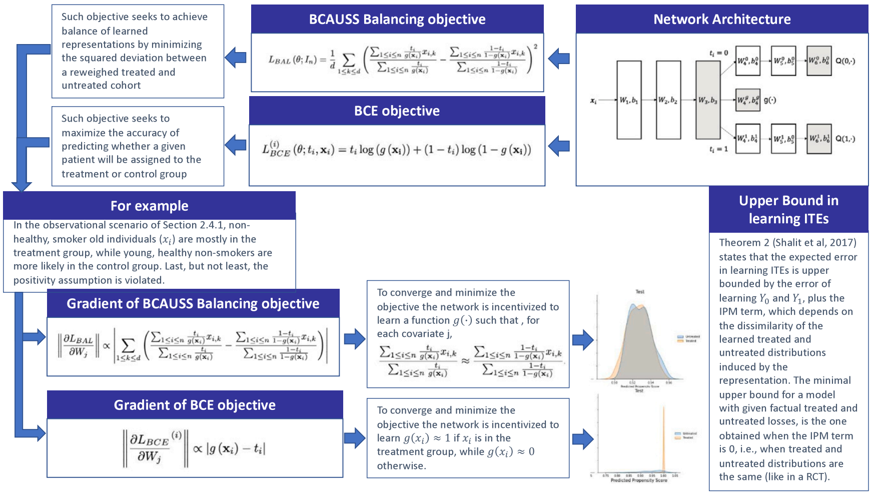
4.1. Analysis of how our method works

Our method offers two straightforward improvements over the existing state-of-the-art methods using binary cross entropy based propensity score estimators. First, our approach is designed to achieve minimal dissimilarity in learning treated and untreated representations, mimicking the requirements of balanced randomization of RCTs, by adopting a specific auto-balancing self-supervised objective. For example, within a real-world data population, the older, sicker, and smokers may be more abundant in the group receiving a particular treatment whereas the younger, healthier, and non-smokers may be more abundant in the control group. What our method does is to learn the representations of both cohorts in which a randomly sampled learned representation from the control cohort will be as similar as possible to a randomly sampled representation from the treated cohort. We show in this section that this balancing effect is crucial when treated and untreated covariate distributions are quite dissimilar and the positivity assumption is violated, which is often the case in real world data [27,28]. In contrast, models that estimate treatment propensity using binary cross-entropy seek to predict which treatment arm each patient was assigned to, learning quite dissimilar treated and untreated distributions when, as we show in this section, the positivity assumption is violated. The second improvement that our method offers is that it induces equal weighting to the gradients of the treatment and control cohorts, ensuring that the learned representations will not be dominated by whichever cohort happens to be larger, which is what happens with binary cross entropy estimators. The rest of this section explains these points in more details.

In situations such as the RCT scenario introduced in Section 2.5.1 where covariates are balanced on treated and control arms, both BCAUSS and binary cross-entropy based methods correctly learn balanced distributions in the representation space, and therefore provide accurate treatment effect estimate. In fact, in such a balanced scenario, $g(\cdot)$ in both BCAUSS and binary cross-entropy based methods can be seen to estimate the true propensity scores after learning. Because all covariates are already identically distributed on both arms, to minimize the objective both models learn the identity transformation, and the distribution of scores produced by $g(\cdot)$ in both models is concentrated around a single point 0.5, and is the same for both treated and control arms. If BCAUSS learns $g(\cdot)$ concentrated around 0.5 for both treated and control minimizes L_{BAL} (Eq. (2)), and can be interpreted as the true propensity score in this RCT scenario. Similarly, also for $g(\cdot)$ trained with binary cross-entropy, the network will learn $g(\cdot)$ concentrated around 0.5 for both treated and control, and adopting a batch size large enough to be representative of the underlying covariates' distribution, will minimize L_{BCE} (Eq. (5)) with $g(\cdot)$ correctly estimating the true propensity score. Hence, also with the binary cross-entropy objective, $g(\cdot)$ learns the true propensity score, and produces the same distribution for treated and control arms, similarly to what happens in our method. This is also evident empirically: in Table 3 in the RCT scenario both our method and the ones adopting binary cross-entropy learn similar treated and untreated distributions.

On the other hand, in the observational scenario introduced in Section 2.5.1, age, comorbidity index, and smoking status have very different distributions between the two arms. While BCAUSS seeks to achieve balance of learned representations by minimizing the squared deviation between a reweighed treated and untreated cohort, binary cross-entropy based methods seek to maximize the accuracy of predicting whether a given patient will be assigned to the treatment or control group. Indeed, the observational scenario of Section 2.5.1 is designed such that older, diabetic, smoking individuals are mostly in the treatment group, while younger, healthier, non-smoking individuals are more likely in the control group. Fig. 5 summarizes this reasoning and depicts, as a result, more similar treated and untreated representations learned by BCAUSS compared to binary cross-entropy based methods which, based on Theorem 2, implies a lower upper bound of the related expected error in learning ITEs. To understand why this happens, see Section 2.4.

In general, if the positivity assumption is satisfied, the binary cross-entropy objective and the BCAUSS objective will tend to learn the same function $g(\cdot)$, i.e., our balancing score is the propensity score after the network is trained because the propensity score minimizes our self-supervised auto-balancing objective (see Rosenbaum [33]). Again, this explains why in the RCT scenario in Table 3, both our method and the ones adopting binary cross-entropy learn similar treated and untreated distributions. However, when the positivity assumption is not satisfied, the binary cross-entropy objective and our self-supervised auto-balancing objective learn very different functions. To converge and minimize the objective, BCAUSS is incentivized to learn a function



Such objective seeks to achieve balance of learned representations by minimizing the squared deviation between a reweighed treated and untreated cohort

BCAUSS Balancing objective

$$L_{BAL}(\theta; I_n) = \frac{1}{d}\sum_{1\leq k\leq d}\left(\frac{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}} - \frac{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}}\right)^2$$

Such objective seeks to maximize the accuracy of predicting whether a given patient will be assigned to the treatment or control group

BCE objective

$$L^{(i)}_{BCE}(\theta; t_i, \mathbf{x}_i) = t_i \log(g(\mathbf{x}_i)) + (1-t_i)\log(1-g(\mathbf{x}_i))$$

Network Architecture

For example

In the observational scenario of Section 2.4.1, non-healthy, smoker old individuals (x_i) are mostly in the treatment group, while young, healthy non-smokers are more likely in the control group. Last, but not least, the positivity assumption is violated.

Gradient of BCAUSS Balancing objective

$$\left\|\frac{\partial L_{BAL}}{\partial W_j}\right\| \propto \left|\sum_{1\leq k\leq d}\left(\frac{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}} - \frac{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}}\right)\right|$$

To converge and minimize the objective the network is incentivized to learn a function $g(\cdot)$ such that , for each covariate j,

$$\frac{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}} \approx \frac{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}}.$$

Gradient of BCE objective

$$\left\|\frac{\partial L_{BCE}{}^{(i)}}{\partial W_j}\right\| \propto |g(\mathbf{x}_i) - t_i|$$

To converge and minimize the objective the network is incentivized to learn $g(x_i) \approx 1$ if x_i is in the treatment group, while $g(x_i) \approx 0$ otherwise.

Upper Bound in learning ITEs

Theorem 2 (Shalit et al, 2017) states that the expected error in learning ITEs is upper bounded by the error of learning Y_0 and Y_1, plus the IPM term, which depends on the dissimilarity of the learned treated and untreated distributions induced by the representation. The minimal upper bound for a model with given factual treated and untreated losses, is the one obtained when the IPM term is 0, i.e., when treated and untreated distributions are the same (like in a RCT).

Fig. 5. Graphical explanation of how the BCAUSS balancing objective works compared to the binary cross-entropy objective. While the former seeks to achieve balance of learned representations by minimizing the squared deviation between a reweighed treated and untreated cohort, the latter seeks to maximize the accuracy of predicting whether a given patient will be assigned to the treatment or control group. For example, the observational scenario of Section 2.5.1 is designed such that non-healthy, smoker old individuals are mostly in the treatment group, while young, healthy non-smokers are more likely in the control group (and the positivity assumption is violated). Hence, to converge and minimize the objective, BCAUSS is incentivized to learn a function $g(\cdot)$ such that, for each covariate k, $\frac{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}} \approx \frac{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}}$. On the contrary, to converge and minimize the objective, BCE-based methods are incentivized to learn $g(\mathbf{x_i}) \approx 1$ if $\mathbf{x_i}$ belongs to the treatment group, and $g(\mathbf{x_i}) \approx 0$ otherwise.

$g(\cdot)$ such that, for each covariate k, $\frac{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{t_i}{g(\mathbf{x}_i)}} \approx \frac{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}x_{i,k}}{\sum_{1\leq i\leq n}\frac{1-t_i}{1-g(\mathbf{x}_i)}}$. On the contrary, to converge and minimize the objective, binary cross-entropy based methods are incentivized to learn $g(\mathbf{x}_i) \approx 1$, if $\mathbf{x}_i$ belongs to the treatment group, and $g(\mathbf{x}_i) \approx 0$, if $\mathbf{x}_i$ belongs to the control group. In regions where the positivity does not hold, such an estimator can be very confident, assuming values close to either one or zero. Such values do not minimize our auto-balancing self-supervised objective, but lead to different treated and untreated learned distributions. On the contrary, even with the positivity violated, BCAUSS still seeks to achieve balance of the learned representations by weighting treated and untreated patients such that the difference between their reweighed averages is much closer, as illustrated in Fig. 3. Specifically, untreated units, mostly lying between 20 and 40 years old, are weighted with weight $\frac{1}{1-g(\mathbf{x}_k)}$ leading to a reweighed average of ≈ 40 instead of 30, while treated units, mostly lying between 40 and 60 years old, are weighted with weight $\frac{1}{g(\mathbf{x}_k)}$ leading to a reweighed average of ≈ 46 instead of 50, reducing L_{BAL} and learning more similar treated and untreated representations, as shown on the right of the figure. Hence, the more similar representations imply a lower upper bound of the related expected error in learning ITEs (Theorem 2).

Further, observational datasets are typically unbalanced in their sizes, e.g., in the illustrative example introduced in Section 2.5.1, the treatment group is 413 times larger than the control group. In this case, when updating the parameters of the representation of the network with SGD, the gradient with respect to the binary cross-entropy objective (see Eq. (20)) can be dominated by the predominant group, making the component related to the other cohort almost negligible and introducing bias in the causal treatment effect estimate. In contrast, such size imbalance is handled well with our model, because it rescales the treated and untreated weighted means (see Eq. (18)). This is evident also empirically: in Table 3 while in the RCT scenario both our method and the ones adopting binary cross-entropy have treated and untreated distributions centered around 0.54 and 0.49, respectively, in the observational scenario, the treated and untreated distributions for methods adopting binary cross-entropy are centered around 0.99, which is a direct effect of the high degree of imbalance of this dataset, showing that the binary cross-entropy objective is not able to mitigate. In contrast, with our model, the treated and untreated distributions are evenly pulled to the middle around 0.5, indicating the two groups equally affect the network training regardless of their sizes.

4.2. *Comparison between treated and untreated distributions induced by the learned representation on IHDP*

In Table 4, on both train-set and test-set, for Dragonnet we have KS test with p-value < 1%, while for BCAUSS we do not. Wasserstein distance is one order of magnitude lower on train-set and two order of magnitudes lower on test-set for BCAUSS compared to Dragonnet. MMD measures are several order of magnitudes lower for BCAUSS compared to Dragonnet, both on train-set and test-set. Furthermore, we notice that the variance of the distribution induced by the learned representation of BCAUSS is one order of magnitude lower, consistent to how covariate distributions should be in RCTs. For example, on test-set for BCAUSS the standard deviation on treated is 0.0128 vs. 0.0064 on untreated, while for Dragonnet, the standard deviation on treated is 0.1137 vs. 0.1204 on untreated.

Table 5 shows the same comparison averaging 1,000 experiments of the IHDP dataset, confirming that treated and untreated distributions learned by BCAUSS are less dissimilar than the ones learned by Dragonnet.

4.3. *Comparison between treated and untreated distributions induced by the learned representation on jobs*

Table 6 shows the comparison between the distributions of treated vs. untreated induced by the learned representation on Dragonnet and BCAUSS on the Jobs dataset, where we confirm what was already observed on the IHDP dataset, i.e.,treated and untreated distributions learned by BCAUSS are more similar than the ones learned by Dragonnet. Specifically, Wasserstein distance is one order of magnitude lower on train-set and test-set for BCAUSS compared to Dragonnet. MMD with the considered kernels are, at least, one order of magnitude lower for BCAUSS compared to Dragonnet, both on train-set and test-set, except for the polynomial kernel where they are the same order of magnitude.

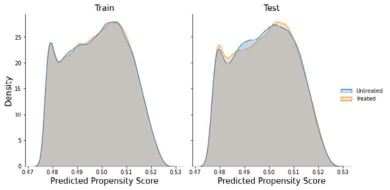
Hence, adopting the auto-balancing self-supervised objective of Eq. (2), the network learns similar representations of the treated and untreated groups, thus enabling less biased estimate of causal treatment effect compared to other approaches such as Dragonnet.

Table 3
Comparison between treated vs. untreated distributions induced by the learned representation of methods adopting binary cross-entropy as propensity objective (BCE) and BCAUSS on the illustrative example introduced in Section 2.5.1. While in the RCT scenario both our method and the ones adopting binary cross-entropy have treated and untreated distributions centered ≈ 0.54 and 0.49, respectively, in the observational scenario the means of the treated and the untreated distributions for methods adopting binary cross-entropy are centered ≈ 0.99, showing that such propensity objective is not able to mitigate the high degree of imbalance of this dataset.

BCE | **BCAUSS**

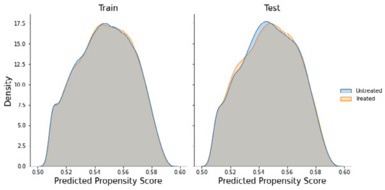
RCT

BCE



	Train	Test
Mean (Tr/Unt.)	0.49/0.49	0.49/0.49
Wasserstein dist.	4.1E-5	4.1E-5
MMD (linear)	7.4E-09	7.4E-09
MMD (rbf)	1.4E-09	1.4E-09
MMD (poly)	7.2E-09	7.2E-09

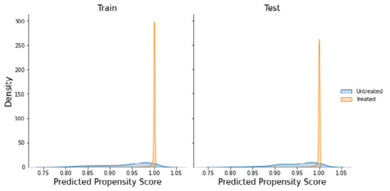
BCAUSS



	Train	Test
Mean (Tr/Unt.)	0.54/0.54	0.54/0.54
Wasserstein dist.	6.82E-05	6.82E-05
MMD (linear)	1.8E-09	1.8E-09
MMD (rbf)	3.6E-09	3.6E-09
MMD (poly)	2.1E-09	2.1E-09

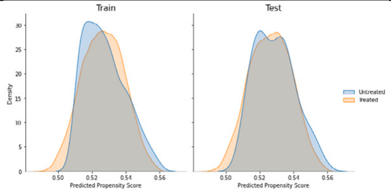
OBS

BCE



	Train	Test
Mean (Tr/Unt.)	0.99/0.99	0.99/0.99
Wasserstein dist.	0.0047	0.0041
MMD (linear)	2.2E-05	2.2E-05
MMD (rbf)	4.4E-05	4.4E-05
MMD (poly)	8.7E-05	8.7E-05

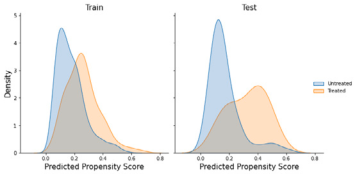
BCAUSS



	Train	Test
Mean (Tr/Unt.)	0.56/0.53	0.56/0.53
Wasserstein dist.	0.0015	0.0022
MMD (linear)	9.5E-07	9.5E-07
MMD (rbf)	1.8E-06	1.8E-06
MMD (poly)	1.0E-06	1.0E-06

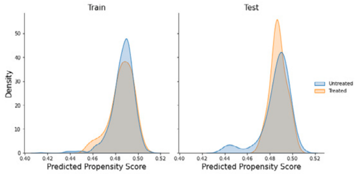
Table 4
Comparison between the treated and untreated distributions induced by the learned representation on Dragonnet and BCAUSS from one experiment of IHDP dataset. Prior covariate treated and control groups have KS test with p-value < 1%, Wasserstein distance 0.1045, MMD(linear) 0.0108, MMD(rbf) 0.0203, MMD(poly) is 0.0023. Given the significance level of 0.01, we can reject the null hypothesis that the two samples were drawn from the same distribution for Dragonnet both on train-set and test-set, while for BCAUSS we cannot. Wasserstein distance is one order of magnitude lower on train-set and two order of magnitudes lower on test-set for BCAUSS compared to Dragonnet. MMD measures are several order of magnitudes lower for BCAUSS compared to Dragonnet, both on train-set and test-set.

Dragonnet



	Train	Test
KS test	0.37	0.59
p-value	1.3E-13	0.0071
Wasserstein dist.	0.0826	0.1583
MMD (linear)	0.0068	0.0238
MMD (rbf)	0.0127	0.0453
MMD (poly)	0.0015	0.0062

BCAUSS

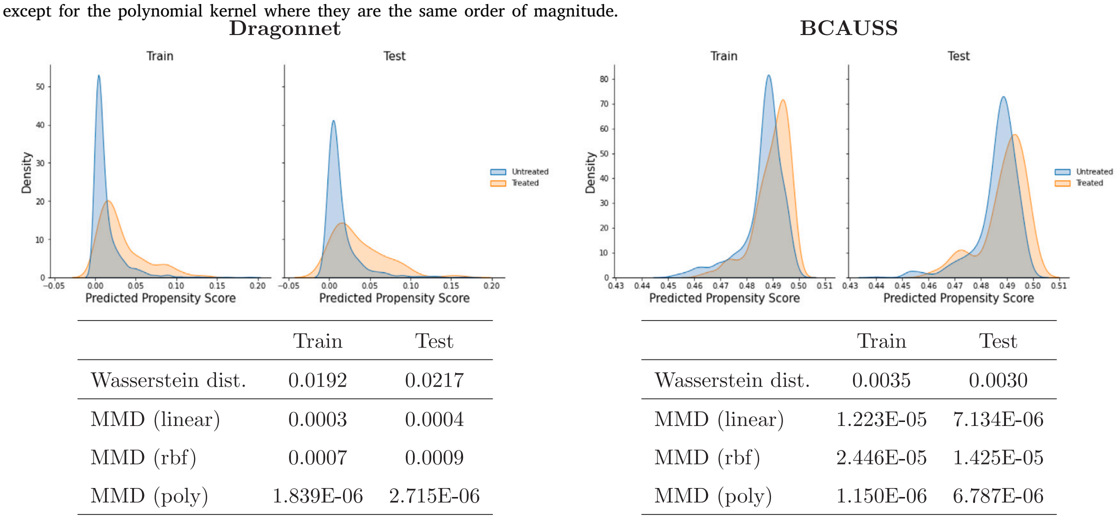


	Train	Test
KS test	0.0851	0.2574
p-value	0.4026	0.6321
Wasserstein dist.	0.0016	0.0042
MMD (linear)	5.9E-07	3.5E-06
MMD (rbf)	1.1E-06	7.1E-06
MMD (poly)	5.6E-07	2.9E-06

Table 5
Comparison between the distributions of treated vs. untreated induced by the learned representation on Dragonnet and BCAUSS averaging 1,000 experiments of the IHDP dataset. Given the significance level of 0.01, on average, we can reject the null hypothesis that the two samples were drawn from the same distribution significantly more often for Dragonnet than BCAUSS both on train-set and test-set. Wasserstein distance is one order of magnitude lower on train-set and test-set for BCAUSS compared to Dragonnet. MMD measures are several order of magnitudes lower for BCAUSS compared to Dragonnet, both on train-set and test-set.

	Dragonnet		BCAUSS	
	Train	Test	Train	Test
KS (*p*-value < 1%)	100%	20.5%	9.9%	1.7%
Wasserstein dist.	0.0940	0.0712	0.0020	0.0043
MMD (linear)	0.0090	0.0051	3.900E−06	1.584E−05
MMD (rbf)	0.0168	0.0097	7.795E−06	3.167E−05
MMD (poly)	0.0020	0.0011	3.809E−06	1.580E−05

Table 6
Comparison between the distributions of treated vs. untreated induced by the learned representation on Dragonnet and BCAUSS from one experiment of Jobs dataset. Wasserstein distance is one order of magnitude lower on train-set and test-set for BCAUSS compared to Dragonnet. MMD with the considered kernels are, at least, one order of magnitude lower for BCAUSS compared to Dragonnet, both on train-set and test-set, except for the polynomial kernel where they are the same order of magnitude.



	Train	Test
Wasserstein dist.	0.0192	0.0217
MMD (linear)	0.0003	0.0004
MMD (rbf)	0.0007	0.0009
MMD (poly)	1.839E-06	2.715E-06

	Train	Test
Wasserstein dist.	0.0035	0.0030
MMD (linear)	1.223E-05	7.134E-06
MMD (rbf)	2.446E-05	1.425E-05
MMD (poly)	1.150E-06	6.787E-06

4.4. Limitations

Our study has several limitations. First, our method has not been validated on a real-world clinical dataset. However, real-world clinical datasets lack counterfactuals, which are necessary to compute the ground truth of ATEs, ATTs, and PEHEs. Without ground truths, evaluating the performance of causal methods is difficult and the related conclusions can be unreliable. We therefore validated our method on Jobs and IHDP datasets, which are widely used for model validation in the literature [8,35,37]. Second, our method is only available for binary treatments as opposed to multiple treatments. In the future, we hope to extend our work to include multiple treatments.

5. Conclusion

We introduced BCAUSS, a multi-task deep neural network able to produce less biased causal estimates from observational data than previously published state-of-the-art methods, thanks to the adoption of an auto-balancing self-supervised objective. Such an objective was specifically designed to learn minimally dissimilar distributions between treated and untreated populations. We explained through ablation analysis, intuitive examples, and equations that the error reduction in causal estimates can be indeed attributed to the balanced representations learned by BCAUSS. In particular, we showed that such a balancing effect is crucial when treated and untreated covariate distributions are quite dissimilar and the positivity assumption is violated, which is often the case in real world data. It is under these circumstances that BCAUSS performs significantly better than the previous state-of-the-art methods. Additionally, unlikely previous state-of-the-arts, BCAUSS induces equal weighting to the gradients of the treatment and control cohorts, ensuring that the learned representations will not be dominated by whichever cohort happens to be larger.

Our method can be applied to cases where class imbalance is severe and covariate distributions makes a comparison of causal effects extremely confounded and unreliable. Beyond the datasets used in this study, our method is applicable to other large observational datasets from EHRs or claims data to provide accurate causal treatment effect estimations, enabling the personalization of treatment recommendations for complex chronic conditions.

CRediT authorship contribution statement

Gino Tesei: Conceptualization, Methodology, Software, Formal analysis, Investigation, Writing – original draft, Writing – review & editing, Visualization. **Stefanos Giampanis:** Validation, Writing – review & editing, Supervision. **Jingpu Shi:** Validation, Investigation, Writing – review & editing. **Beau Norgeot:** Resources, Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Gino Tesei reports a relationship with Elevance Health Inc that includes: employment and equity or stocks. Stefanos Giampanis reports a relationship with Elevance Health Inc that includes: employment and equity or stocks. Jingpu Shi reports a relationship with Elevance Health Inc that includes: employment and equity or stocks. Beau Norgeot reports a relationship with Elevance Health Inc that includes: employment and equity or stocks.

Acknowledgments

We would like to thank Ben Glicksberg for his valuable advice and Daniel Brown for his helpful discussions. This work was supported by Elevance Health, USA.

Reproducibility statement

To contribute to the methods' reproducibility, we provide the code for replicating experiments and Jupyter Notebooks used to perform our analysis. Specifically, the repository is available at https://github.com/anthem-ai/bcauss. In Section 2.5 we report the links to download benchmarks and real-world datasets for replicating experiments and we describe the data processing steps referencing the original papers where such datasets were first introduced. Also, in Section 2.6 the evaluation metrics for each dataset are defined and their calculations are explained.

Appendix A. Back-propagating the gradient with respect to the auto-balancing objective

The parameters of the network of Fig. 1 are updated at each iteration to minimize $J\left(\theta; P_n\right)$, which in case of batch gradient descent [61] means

$$\theta^{(i+1)} := \theta^{(i)} - \alpha \frac{\partial J}{\partial \theta} = \theta^{(i)} - \alpha \left(\lambda_{BAL} \frac{\partial L_{BAL}}{\partial \theta} + \frac{1}{n} \sum_{1 \le j \le n} \frac{\partial L_{REG}^{(j)}}{\partial \theta} \right), \tag{9}$$

where α is the learning rate. Hence, regarding the factual and counterfactual losses, using chain rule for partial derivatives[1]

$$\frac{\partial L_{REG}^{(j)}}{\partial z_{j,5}^{t_j}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,6}^{t_j}} \frac{\partial z_{j,6}^{t_j}}{\partial a_{j,5}^{t_j}} \frac{\partial a_{j,5}^{t_j}}{\partial z_{j,5}^{t_j}} = 2 \left[y^{(j)} - Q\left(t_j, \mathbf{x}_j\right) \right] W_6^{t_j} \odot H\left(z_{j,5}^{t_j}\right) \tag{10}$$

$$\frac{\partial L_{REG}^{(j)}}{\partial z_{j,4}^{t_j}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,5}^{t_j}} \frac{\partial z_{j,5}^{t_j}}{\partial a_{j,4}^{t_j}} \frac{\partial a_{j,4}^{t_j}}{\partial z_{j,4}^{t_j}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,5}^{t_j}} W_5^{t_j} \odot H\left(z_{j,4}^{t_j}\right) \tag{11}$$

$$\frac{\partial L_{REG}^{(j)}}{\partial z_{j,3}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,4}^{t_j}} \frac{\partial z_{j,4}^{t_j}}{\partial a_{j,3}} \frac{\partial a_{j,3}}{\partial z_{j,3}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,4}^{t_j}} W_4^{t_j} \odot H\left(z_{j,3}\right) \tag{12}$$

$$\frac{\partial L_{REG}^{(j)}}{\partial z_{j,2}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,3}} \frac{\partial z_{j,3}}{\partial a_{j,2}} \frac{\partial a_{j,2}}{\partial z_{j,2}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,3}} W_3 \odot H\left(z_{j,2}\right) \tag{13}$$

$$\frac{\partial L_{REG}^{(j)}}{\partial z_{j,1}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,2}} \frac{\partial z_{j,2}}{\partial a_{j,1}} \frac{\partial a_{j,1}}{\partial z_{j,1}} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,2}} W_2 \odot H\left(z_{j,1}\right) \tag{14}$$

where $H(\cdot)$ is the Heaviside step function. In order to compute, for example, $\frac{\partial L_{REG}^{(j)}}{\partial W_1}$, we have

$$\delta_{1,j} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,1}} \tag{15}$$

$$\frac{\partial L_{REG}^{(j)}}{\partial W_1} = \frac{\partial L_{REG}^{(j)}}{\partial z_{j,1}} \frac{\partial z_{j,1}}{\partial W_1} = \delta_{1,j}^T \mathbf{x}_j^T \tag{16}$$

In Eq. (16) it has been used the practice (in a slight abuse of notation) of making the Jacobian $\frac{\partial L_{REG}^{(j)}}{\partial W_1}$ of the same shape as W_1. This is a well consolidated practice in deep learning literature, since this matrix has the same shape as W_1 we could just subtract it (times the learning rate) from W_1 when doing gradient descent. Relationships for other matrices and bias terms can be derived with similar reasoning. In the same way, regarding the auto-balancing loss, applying the product rule and the chain rule, we have

$$\frac{\partial L_{BAL}}{\partial W_1} = \frac{2}{d} \sum_{1 \le j \le d} \left\{ f_j\left(g, I_n\right) \sum_{1 \le i \le n} \frac{\partial f_j}{\partial g_i} \frac{\partial g_i}{\partial z_i^g} \frac{\partial z_i^g}{\partial z_{i,3}} \frac{\partial z_{i,3}}{\partial z_{i,2}} \frac{\partial z_{i,2}}{\partial z_{i,1}} \frac{\partial z_{i,1}}{\partial W_1} \right\} \tag{17}$$

$$= \frac{2}{d} \sum_{1 \le j \le d} \left\{ f_j\left(g, I_n\right) \sum_{1 \le i \le n} \frac{\partial f_j}{\partial g_i} \sigma\left(z_i^g\right) \left[1 - \sigma\left(z_i^g\right)\right] \gamma_i^T \mathbf{x}_\mathbf{i}^\mathbf{T} \right\} \tag{18}$$

where

$$f_j\left(g, I_n\right) = \left(\frac{\sum_{1 \le i \le n} \frac{t_i}{g(\mathbf{x}_i)} x_{i,j}}{\sum_{1 \le i \le n} \frac{t_i}{g(\mathbf{x}_i)}} - \frac{\sum_{1 \le i \le n} \frac{1-t_i}{1-g(\mathbf{x}_i)} x_{i,j}}{\sum_{1 \le i \le n} \frac{1-t_i}{1-g(\mathbf{x}_i)}} \right),$$

$$\gamma_i = \left[W_4^q \odot H\left(z_{i,3}\right) W_3 \odot H\left(z_{i,2}\right) W_2 \odot H\left(z_{i,1}\right) \right],$$

and it is used the derivative of the sigmoid function, i.e., $\frac{\partial \sigma(a)}{\partial a} = \sigma(a)\left[1 - \sigma(a)\right]$. Relationships for other matrices and bias terms can be derived with similar reasoning. When we plug Eq. (16) and (18) into (9) for each matrix and bias term of the network, we can understand better how backpropagation works on BCAUSS. The term of Eq. (10) is a standard regression term depending on the difference between observed and predicted outcome. This term is the same for networks like Dragonnet and TARNet. The term of Eq. (18) is our adjustment due to the dissimilarity between learned treated and untreated distributions. Specifically, if there is no dissimilarity between learned treated and untreated distributions, then this term is zero and the causal problem degenerates into a standard regression problem, as it should be. On the other hand, if there is dissimilarity between learned treated and untreated distributions, i.e., $\exists j \in \{1, \dots, d\}$ so that $f_j\left(g, I_n\right) \neq 0$, then the updates of the network parameters are adjusted with the term of Eq. (18), which depends on $f_j\left(g, I_n\right)$ and the derivative of $f_j\left(g, I_n\right)$ with respect to the predicted propensity scores of each observation. Now, we can understand better why SGD with large batch-size works better than SGD with small batch-size and even Adam. For example, if for Adam for a given j we have that $f_j\left(g, I_n\right) = 0$ but $f_j\left(g, I_k\right) \neq 0$, where k is the batch-size ($k < n$), we update our parameters with the exponentially weighted average of the terms of Eq. (18). These updates are compensated with the updates of the following batches (as $f_j\left(g, I_n\right) = 0$ and $f_j\left(g, I_k\right) \neq 0$) not linearly but in a exponentially weighted averaging way leading, in general, to sub-optimal states of the network. In other words, if $f_j\left(g, I_n\right) = 0$ then the learned representations of the network are already similar enough at least for the covariate j and there is no need of adjustments during the backpropagation, and this happens only adopting SGD and a large batch size (in this example, the whole train set). Further, if the batch size is too small, we can observe numerical problems for SGD due to specific values of $f_j\left(g, I_k\right)$, while such values are smoothed adopting Adam.

Appendix B. Back-propagating the gradient with respect to the binary cross-entropy objective

A different objective used in the literature to train the propensity score estimator is the binary cross-entropy objective of Eq. (5). Even Rosenbaum and Rubin [11] claim that the propensity score "*may be estimated from observed data, perhaps using a model such as a logit model*". We find interesting such use of "*perhaps*" in the previous sentence. Maybe, what the authors mean is that for estimating the propensity score adopting a logit model, we need a train-sample very

[1] If $f : \mathbb{R} \rightarrow \mathbb{R}$ is a differentiable function applied element-wise to a vector $\mathbf{x}$, i.e., $z = f(\mathbf{x}) = \left[f\left(x_1\right), f\left(x_2\right), \dots, f\left(x_d\right)\right]^T$, then it is possible to show that the Jacobian $\frac{\partial z}{\partial \mathbf{x}} = diag\left(f'(\mathbf{x})\right) = \begin{pmatrix} \frac{\partial f(x_1)}{\partial x_1} & 0 & \cdots & 0 \\ 0 & \frac{\partial f(x_2)}{\partial x_2} & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \frac{\partial f(x_d)}{\partial x_d} \end{pmatrix}$. Since multiplication by a diagonal matrix is the same as doing element-wise multiplication by the diagonal, we could also write $\odot f'(\mathbf{x})$ when applying the chain rule.

representative of the underlying covariate distribution. With this in mind let analyze the effect of back-propagating the gradient with respect to the binary cross-entropy objective. Assuming batch gradient descent [61], the parameters of the network are updated at each iteration according to the following optimization step

$$\theta^{(i+1)} := \theta^{(i)} - \alpha \frac{\partial J}{\partial \theta} = \theta^{(i)} - \frac{\alpha}{n}\left[\sum_{1\leq j\leq n}\left(\frac{\partial L_{REG}}{\partial \theta}^{(j)} + \lambda_{BCE}\frac{\partial L_{BCE}}{\partial \theta}^{(j)}\right)\right], \tag{19}$$

where α is the learning rate. The components $\frac{\partial L_{REG}}{\partial \theta}$ can be derived like for the auto-balancing objective, while for components $\frac{\partial L_{BCE}}{\partial \theta}$ we have

$$\frac{\partial L_{BCE}}{\partial W_1}^{(j)} = \left\{\left[g\left(\mathbf{x_j}\right) - t_j\right] \mathbf{W}_4^g \odot H\left(z_{j,3}\right) W_3 \odot H\left(z_{j,2}\right) W_2 \odot H\left(z_{j,1}\right)\right\}^T \mathbf{x_j^T} \tag{20}$$

Relationships for other matrices and bias terms can be derived with similar reasoning. The term of Eq. (20) is the adjustment (times λ_{BCE}) for the gradient update $\frac{\partial L_{REG}}{\partial \theta}^{(j)}$ depending on the difference of the decision of treatment of the jth observation and the related predicted propensity score. Hence, we can understand better why working with large batch-size and SGD is beneficial also for the binary cross-entropy objective, as recalled already in [10]. The reason is that a larger batch size should be more representative of the underlying covariate distribution compared to a smaller batch size and, hence, we should have an higher chance to learn the true propensity score. For example, if the true propensity score for a given $\mathbf{x_j} \in \mathcal{X}$ is $1/3$ but in the current batch there is only one observation belonging to the treated group having covariate $\mathbf{x_j}$, then we update the parameters of the network with a component $\left\{\left[g\left(\mathbf{x_j}\right) - t_j\right] \mathbf{W}_4^g \odot H\left(z_{j,3}\right) W_3 \odot H\left(z_{j,2}\right) W_2 \odot H\left(z_{j,1}\right)\right\}^T \mathbf{x_j^T}$ which can be zero, in general, only in case $g\left(\mathbf{x_j}\right) = 1$, which is the wrong value for $g\left(\mathbf{x_j}\right)$ to converge. Ideally, at each optimization step, we need a batch-size large enough to represent the underlying covariate distribution. Indeed, in our experiments we noticed improvements in models adopting the binary cross-entropy objective with large batch-sizes. As final step, let assume we adopt the maximum possible batch-size and let ask what are the implications for the network if we adopt the binary cross-entropy objective. The network will learn a propensity score estimator such that $g(\mathbf{x}) \approx p_\theta(t = 1\,|\mathbf{x})$ which, if the positivity assumption is not satisfied, pushes the network to learn dissimilar treated and untreated distributions as explained in Section 4.1.

Glossary

activation function An activation function in a neural network defines how the weighted sum of the inputs is transformed into an output from a node of the network. Typically, activation functions are non-linear (e.g., ReLU, ELU, Tanh) and there is a theoretical reason for this. Indeed, the universal approximation theorem (Hornik et al. [62]; Cybenko [63]) states that a feed-forward neural network with a linear output layer and at least one hidden layer with any “squashing” non-linear activation function (e.g., Tanh) can approximate any Borel measurable function from one finite-dimensional space to another with any desired nonzero amount of error, provided that the network is given enough hidden units.

Adam Algorithm for first-order gradient-based optimization of stochastic objective functions, based on adaptive estimates of lower-order moments. In contrast to SGDs, Adam uses *momentum*, i.e.,a rolling average of the gradients, to reduce the variance of the updates of the network. Further, it adopts *adaptive learning rates*, i.e.,divides the final network update by the weighted average of the (element-wise) squares of the magnitudes of the gradients, which has a further regularization effect. Overall, Adam is appropriate for non-stationary objectives and problems with very noisy and/or sparse gradients (Kingma and Ba [64]).

backpropagation A widely used algorithm for training neural networks. In fitting a neural network, backpropagation computes the gradient of the loss function with respect to the weights of the network very efficiently. This efficiency makes it feasible to use gradient methods (e.g., SGD or Adam) for training multilayer networks, updating weights to minimize loss. The backpropagation algorithm works by computing the gradient of the loss function with respect to each weight by the chain rule, computing the gradient one layer at a time, iterating backward from the last layer to avoid redundant calculations of intermediate terms in the chain rule (dynamic programming).

batch gradient descent Iterative algorithm for optimizing objective functions that uses all available data to form an accurate expectation of the gradient (Ruder [61]).

binary cross-entropy A loss function that is used in binary classification tasks, i.e.,tasks that answer a question with only two choices. Formally, it is the negative average of the log-corrected predicted probabilities (e.g., see Eq. (5)). The lower the value, the better. In causal inference methods based on neural networks, it is typically used to train the propensity score estimator.

ELU The Exponential Linear Unit (ELU) (Clevert et al. [65]) is an activation function for neural networks defined as

$$f(x) = \begin{cases} x & if\ x > 0 \\ \alpha\left(\exp(x) - 1\right) & if\ x \leq 0 \end{cases}$$

. with $\alpha > 0$. In contrast to ReLUs, ELUs come with non-zero gradients for negative arguments, which prevents vanishing gradients when the function works in this region. However, it is computationally more expensive than ReLU (e.g., exponential operator) and require further parameters to learn (e.g., α), increasing the capacity of the network and making the network more prone to overfitting.

gradient The gradient of a scalar-valued differentiable function f of several variables is the vector field ∇f whose value at a point p is the vector with components the partial derivatives of f at p. That is, for $f : \mathbb{R}^n \to \mathbb{R}$, its gradient $\nabla f : \mathbb{R}^n \to \mathbb{R}^n$ is defined at the point p as $\left[\frac{\partial f}{\partial x_1}(p), \ldots, \frac{\partial f}{\partial x_n}(p)\right]^T$.

Heaviside step function Function, named after Oliver Heaviside (1850–1925), the value of which is zero for negative arguments and one for positive arguments.

Jacobian The Jacobian matrix of a vector-valued function of several variables is the matrix of all its first-order partial derivatives. That is, for $f : \mathbb{R}^n \to \mathbb{R}^m$, its Jacobian $\frac{\partial \mathbf{f}}{\partial \mathbf{x}} : \mathbb{R}^n \to \mathbb{R}^m$ is defined as

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} := \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \cdots & \ddots & \cdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

.

Kolmogorov–Smirnov Nonparametric test (Kolmogorov [43],Smirnov [44]) of the equality of continuous, one-dimensional probability distributions that can be used to compare a sample with a reference probability distribution (one-sample KS test), or to compare two samples (two-sample KS test).

learned representations Representations learning is a class of machine learning approaches (Bengio et al. [9]) that allow a system to learn the representations of the data that make it easier to extract useful information to perform the predictive task. Representation learning has become a field in itself in the machine learning community, with regular workshops and sometimes under the header of *Deep Learning* or *Feature Learning*.

MMD Maximum Mean Discrepancy (Gretton et al. [31]) is a distance metric on the space of probability measures based on the notion of embedding probabilities in a reproducing kernel Hilbert space.

ReLU Rectified Linear Units (Nair and Hinton [32]) are a type of activation function that are linear for positive arguments, and zero for negative ones, i.e., $f(x) = \max(0, x)$. The peculiarity of this function is the source of non-linearity. Linearity for positive arguments has the attractive property that it prevents vanishing gradients, which is what happens for activation functions like Tanh. For negative arguments the problem remains. This is the reason of its many variations (e.g., LeakyReLU or ELU), that however are computationally more expensive (e.g., exponential operator for ELU) and require further parameters to learn (e.g., α for ELU), increasing the capacity of the network and making the network more prone to overfitting.

SGD Stochastic Gradient Descent is an iterative algorithm for optimizing objective functions that uses mini-batches of data to form an expectation of the gradient, rather than all available data. It can be regarded as a stochastic approximation of batch gradient descent, since it replaces the actual gradient (calculated from the entire data set) by an estimate thereof (calculated from a randomly selected subset of the data). Especially in high-dimensional optimization problems this reduces the very high computational burden, achieving faster iterations in trade for less accurate estimates of the gradient.

Tanh Activation function used for neural networks adopting hyperbolic tangent as non-linearity, i.e., $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

targeted regularization term Regularization term introduced by Shi et al. [10] to achieve asymptotically robustness, taking inspiration from Targeted Learning [7], which adjusts the estimation of an initial statistical model in a step targeted toward making an optimal bias–variance trade-off of the causal effect.

Wasserstein distance Distance function (Villani [41] and Cuturi and Doucet [42]) defined between probability distributions inspired by the problem of optimal mass transportation. Intuitively, each distribution is viewed as an amount of earth (soil) piled, and the metric is the minimum ‘‘cost’’ of turning one pile into the other.

References

[1] N.E. Munk, J.S. Knudsen, A. Pottegård, D.R. Witte, R.W. Thomsen, Differences between randomized clinical trial participants and real-world empagliflozin users and the changes in their glycated hemoglobin levels, JAMA (2020).

[2] D.C. Klonoff, The expanding role of real-world evidence trials in health care decision making, J. Diabetes Sci. Technol. (2020).

[3] A. Belloni, V. Chernozhukov, C. Hansen, Inference on treatment effects after selection among high-dimensional controls†, Rev. Econom. Stud. 81 (2) (2013) 608–650, http://dx.doi.org/10.1093/restud/rdt044.

[4] S. Athey, G.W. Imbens, S. Wager, Approximate residual balancing: De-biased inference of average treatment effects in high dimensions, 2018, arXiv:1604.07125.

[5] V. Chernozhukov, D. Chetverikov, M. Demirer, E. Duflo, C. Hansen, W. Newey, J. Robins, Double/debiased machine learning for treatment and causal parameters, 2017, arXiv:1608.00060.

[6] P. Austin, An introduction to propensity score methods for reducing the effects of confounding in observational studies, Multivar. Behav. Res. 46 (2011) 399–424.

[7] M.J. van der Laan, S. Rose, Targeted Learning in Data Science: Causal Inference for Complex Longitudinal Studies, first ed., Springer Publishing Company, Incorporated, 2018.

[8] U. Shalit, F.D. Johansson, D. Sontag, Estimating individual treatment effect: Generalization bounds and algorithms, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 3076–3085.

[9] Y. Bengio, A.C. Courville, P. Vincent, Representation learning: A review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (2013) 1798–1828.

[10] C. Shi, D.M. Blei, V. Veitch, Adapting neural networks for the estimation of treatment effects, in: H.M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E.B. Fox, R. Garnett (Eds.), NeurIPS, 2019, pp. 2503–2513, URL: http://dblp.uni-trier.de/db/conf/nips/nips2019.html#ShiBV19.

[11] P.R. Rosenbaum, D.B. Rubin, The central role of the propensity score in observational studies for causal effects, Biometrika 70 (1) (1983) 41–55, http://dx.doi.org/10.1093/biomet/70.1.41.

[12] X. Su, C.-L. Tsai, H. Wang, D.M. Nickerson, B. Li, Subgroup analysis via recursive partitioning, J. Mach. Learn. Res. 10 (5) (2009) 141–158, URL: http://jmlr.org/papers/v10/su09a.html.

[13] S. Athey, G. Imbens, Recursive partitioning for heterogeneous causal effects, Proc. Natl. Acad. Sci. 113 (27) (2016) 7353–7360, http://dx.doi.org/10.1073/pnas.1510489113.

[14] W. Zhang, T.D. Le, L. Liu, Z.-H. Zhou, J. Li, Mining heterogeneous causal effects for personalized cancer treatment, in: J. Wren (Ed.), Bioinformatics 33 (15) (2017) 2372–2378, http://dx.doi.org/10.1093/bioinformatics/btx174.

[15] S.R. Künzel, J.S. Sekhon, P.J. Bickel, B. Yu, Metalearners for estimating heterogeneous treatment effects using machine learning, Proc. Natl. Acad. Sci. 116 (10) (2019) 4156–4165, http://dx.doi.org/10.1073/pnas.1804597116.

[16] S. Wager, S. Athey, Estimation and inference of heterogeneous treatment effects using random forests, 2017, arXiv:1510.04342.

[17] A.M. Alaa, M. Weisz, M. van der Schaar, Deep counterfactual networks with propensity-dropout, 2017, arXiv:1706.05966.

[18] P. Schwab, L. Linhardt, W. Karlen, Perfect match: A simple method for learning representations for counterfactual inference with neural networks, 2019, arXiv:1810.00656.

[19] C.W. Belthangady, B. Norgeot, Minimizing bias in massive multi-arm observational studies with BCAUS: Balancing covariates automatically using supervision, BMC Med. Res. Methodol. (2021).

[20] C. Louizos, U. Shalit, J.M. Mooij, D. Sontag, R. Zemel, M. Welling, Causal effect inference with deep latent-variable models, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, URL: https://proceedings.neurips.cc/paper/2017/file/94b5bde6de888ddf9cde6748ad2523d1-Paper.pdf.

[21] J. Yoon, J. Jordon, M. van der Schaar, GANITE: Estimation of individualized treatment effects using generative adversarial nets, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018, URL: https://openreview.net/forum?id=ByKWUeWA-.

[22] A.M. Alaa, M. van der Schaar, Bayesian inference of individualized treatment effects using multi-task Gaussian processes, in: I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 30, Curran Associates, Inc., 2017, URL: https://proceedings.neurips.cc/paper/2017/file/6a508a60aa3bf9510ea6acb021c94b48-Paper.pdf.

[23] D.P. Kingma, M. Welling, Stochastic gradient VB and the variational autoencoder, in: Second International Conference on Learning Representations, Vol. 19, no. 121, ICLR, 2014.

[24] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Vol. 2, NIPS '14, MIT Press, Cambridge, MA, USA, 2014, pp. 2672–2680.

[25] J. Shi, D. Wang, G. Tesei, B. Norgeot, Generating high-fidelity privacy-conscious synthetic patient data for causal effect estimation with multiple treatments, Front. Artif. Intell. 5 (2022) http://dx.doi.org/10.3389/frai.2022.918813, URL: https://www.frontiersin.org/articles/10.3389/frai.2022.918813.

[26] W. Zhang, L. Liu, J. Li, Treatment effect estimation with disentangled latent factors, 2021, arXiv:2001.10652.

[27] Y. Zhu, R.A. Hubbard, J. Chubak, J. Roy, N. Mitra, Core concepts in pharmacoepidemiology: Violations of the positivity assumption in the causal analysis of observational data: Consequences and statistical approaches, Pharmacoepidemiol Drug Saf. 11 (30) (2021) 1471–1485, http://dx.doi.org/10.1002/pds.5338, PMID: 34375473.

[28] M.L. Petersen, K.E. Porter, S. Gruber, Y. Wang, M.J. van der Laan, Diagnosing and responding to violations in the positivity assumption, Stat. Methods Med. Res. 21 (1) (2012) 31–54, http://dx.doi.org/10.1177/0962280210386207, PMID: 21030422.

[29] J. Pearl, Causality: Models, Reasoning and Inference, second ed., Cambridge University Press, USA, 2009.

[30] Sriperumbudur, K. Bharath, K. Fukumizu, A. Gretton, B. Schølkopf, G.R. Lanckriet, et al., On the empirical estimation of integral probability metrics, Electron. J. Stat. (2012).

[31] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. 13 (25) (2012) 723–773, URL: http://jmlr.org/papers/v13/gretton12a.html.

[32] V. Nair, G.E. Hinton, Rectified linear units improve restricted Boltzmann machines, in: Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML '10, Omni Press, Madison, WI, USA, 2010, pp. 807–814.

[33] P.R. Rosenbaum, Model-based direct adjustment, J. Amer. Statist. Assoc. 82 (398) (1987) 387–394, http://dx.doi.org/10.1080/01621459.1987.10478441, URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1987.10478441.

[34] M.E. Charlson, et al., A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation, J. Chronic Dis. (1987).

[35] J.L. Hill, Bayesian nonparametric modeling for causal inference, J. Comput. Graph. Statist. 20 (1) (2011) 217–240, http://dx.doi.org/10.1198/jcgs.2010.08162.

[36] R. LaLonde, Evaluating the econometric evaluations of training programs with experimental data, Amer. Econ. Rev. 76 (4) (1986) 604–620, URL: https://EconPapers.repec.org/RePEc:aea:aecrev:v:76:y:1986:i:4:p:604-20.

[37] R. Dehejia, S. Wahba, Propensity score-matching methods for nonexperimental causal studies, Rev. Econ. Stat. 84 (1) (2002) 151–161, URL: https://EconPapers.repec.org/RePEc:tpr:restat:v:84:y:2002:i:1:p:151-161.

[38] U. Shalit, F.D. Johansson, D. Sontag, Estimating individual treatment effect: generalization bounds and algorithms, in: D. Precup, Y.W. Teh (Eds.), Proceedings of the 34th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 70, PMLR, 2017, pp. 3076–3085, URL: http://proceedings.mlr.press/v70/shalit17a.html.

[39] C. Louizos, U. Shalit, J. Mooij, D. Sontag, R. Zemel, M. Welling, Causal effect inference with deep latent-variable models, in: NIPS, 2017.

[40] L. Yao, S. Li, Y. Li, M. Huai, J. Gao, A. Zhang, Representation learning for treatment effect estimation from observational data, in: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, Inc., 2018, URL: https://proceedings.neurips.cc/paper/2018/file/a50abba8132a77191791390c3eb19fe7-Paper.pdf.

[41] C. Villani, Optimal Transport: Old and New, Vol. 338, Springer Science & Business Media, 2008.

[42] M. Cuturi, A. Doucet, Fast computation of Wasserstein Barycenters, in: E.P. Xing, T. Jebara (Eds.), Proceedings of the 31st International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 32, (no. 2) PMLR, Bejing, China, 2014, pp. 685–693, URL: https://proceedings.mlr.press/v32/cuturi14.html.

[43] A. Kolmogorov, Sulla determinazione empirica di una legge di distribuzione, in: Giornale Dell'Istituto Italiano Degli Attuari, 1933.

[44] N. Smirnov, Table for estimating the goodness of fit of empirical distributions, in: Annals of Mathematical Statistics, 1948.

[45] H. Robbins, A stochastic approximation method, Ann. Math. Stat. 22 (2007) 400–407.

[46] Y. Yao, L. Rosasco, A. Caponnetto, On early stopping in gradient descent learning, Constr. Approx (2007) 289–315.

[47] P.Z. Schochet, Is regression adjustment supported by the Neyman model for causal inference, J. Statist. Plann. Inference 140 (2007) 246–259.

[48] C. Louizos, U. Shalit, J. Mooij, D. Sontag, R. Zemel, M. Welling, Causal effect inference with deep latent-variable models, 2017, arXiv:1705.08821.

[49] F.D. Johansson, U. Shalit, D. Sontag, Learning representations for counterfactual inference, in: Proceedings of the 33rd International Conference on Machine Learning, Vol. 48, 2016, pp. 3020–3029, arXiv:1605.03661.

[50] R.K. Crump, V.J. Hotz, G.W. Imbens, O.A. Mitnik, Nonparametric tests for treatment effect heterogeneity, Rev. Econ. Stat. 90 (3) (2008) 389–405, http://dx.doi.org/10.1162/rest.90.3.389.

[51] H.A. Chipman, E.I. George, R.E. McCulloch, BART: Bayesian additive regression trees, Ann. Appl. Stat. 4 (1) (2010) 266–298, http://dx.doi.org/10.1214/09-AOAS285.

[52] U. Shalit, F.D. Johansson, D. Sontag, Estimating individual treatment effect: Generalization bounds and algorithms, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 3076–3085.

[53] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[54] S. Wager, S. Athey, Estimation and inference of heterogeneous treatment effects using random forests, J. Amer. Statist. Assoc. 113 (523) (2018) 1228–1242, http://dx.doi.org/10.1080/01621459.2017.1319839.

[55] J.M. Robins, A. Rotnitzky, L.P. Zhao, Analysis of semiparametric regression models for repeated outcomes in the presence of missing data, J. Amer. Statist. Assoc. 90 (429) (1995) 106–121, http://dx.doi.org/10.1080/01621459.1995.10476493, URL: https://www.tandfonline.com/doi/abs/10.1080/01621459.1995.10476493.

[56] C.F. Kurz, Augmented inverse probability weighting and the double robustness property, Med. Decis. Mak. 42 (2) (2022) 156–167, http://dx.doi.org/10.1177/0272989X211027181, PMID: 34225519.

[57] C. Belthangady, S. Giampanis, I. Jankovic, W. Stedden, P. Alves, S. Chong, C. Knott, B. Norgeot, Causal deep learning reveals the comparative effectiveness of antihyperglycemic treatments in poorly controlled diabetes, Nature Commun. 13 (1) (2022) 6921, http://dx.doi.org/10.1038/s41467-022-33732-9.

[58] E.J. Williamson, A. Forbes, I.R. White, Variance reduction in randomised trials by inverse probability weighting using the propensity score, Stat. Med. 33 (5) (2014) 721–737.

[59] H. Bang, J.M. Robins, Doubly robust estimation in missing data and causal inference models, Biometrics 61 (4) (2005) 962–973.

[60] J. Robins, M. Sued, Q. Lei-Gomez, A. Rotnitzky, Comment: Performance of double-robust estimators when" inverse probability" weights are highly variable, Statist. Sci. 22 (4) (2007) 544–559.

[61] S. Ruder, An overview of gradient descent optimization algorithms, 2016, arXiv preprint arXiv:1609.04747.

[62] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366, http://dx.doi.org/10.1016/0893-6080(89)90020-8, URL: https://www.sciencedirect.com/science/article/pii/0893608089900208.

[63] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Control Signals Systems 2 (4) (1989) 303–314, http://dx.doi.org/10.1007/BF02551274.

[64] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization (Thesis), 2014, URL: http://arxiv.org/abs/1412.6980, cite arxiv:1412.6980, Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[65] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), in: Y. Bengio, Y. LeCun (Eds.), ICLR (Poster), 2016, URL: http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#ClevertUH15.