Identifiable Energy-based Representations: An Application to Estimating Heterogeneous Causal Effects

Yao Zhang[*]
University of Cambridge

Jeroen Berrevoets[*]
University of Cambridge

Mihaela van der Schaar
University of Cambridge
UCLA
The Alan Turing Institute

Abstract

Conditional average treatment effects (CATEs) allow us to understand the effect heterogeneity across a large population of individuals. However, typical CATE learners assume all confounding variables are measured in order for the CATE to be identifiable. This requirement can be satisfied by collecting many variables, at the expense of increased sample complexity for estimating CATEs. To combat this, we propose an energy-based model (EBM) that learns a low-dimensional representation of the variables by employing a noise contrastive loss function. With our EBM we introduce a preprocessing step that alleviates the dimensionality curse for *any existing* learner developed for estimating CATEs. We prove that our EBM keeps the representations partially identifiable up to some universal constant, as well as having universal approximation capability. These properties enable the representations to converge and keep the CATE estimates consistent. Experiments demonstrate the convergence of the representations, as well as show that estimating CATEs on our representations performs better than on the variables or the representations obtained through other dimensionality reduction methods.

1 Introduction

Average treatment effect (ATE) is arguably the most popular estimand in the causal inference literature.
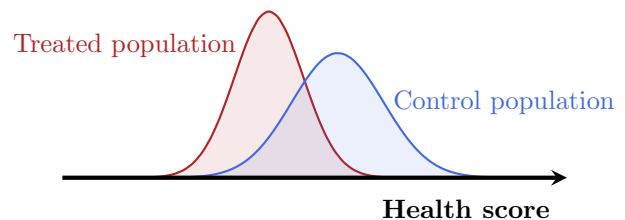

Figure 1: **Imbalanced treated and control (i.e. untreated) populations.** Individuals with lower health scores are more likely to receive the treatment.

With the ATE, one measures if a treatment is effective on average over a population of individuals. However, even if we estimate an ATE accurately, we can not conclude if a treatment is beneficial for a particular individual. In order to get treatment effect estimates for one individual, we condition the ATE on the individual of interest, and arrive at the *conditional* average treatment effect (CATE). CATEs know successful applications in areas such as healthcare and education.

While clinical trials represent the gold standard for causal inference, they often have a small number of individuals and narrow inclusion criteria, rendering them unsuitable for use in estimating the causal effects conditional on some particular individual's confounding variables (covariates). On the other hand, observational datasets are becoming increasingly available, but require careful attention to the biases in the datasets. There is growing interest in leveraging observational data to estimate CATEs, e.g., electronic healthcare records used to determine which patients should get what treatments, or school records to optimize educational policy in low- and high-income communities.

A fundamental assumption for valid causal inference on observational data is called the strong ignorability assumption [Rosenbaum and Rubin, 1983, 1984]. It assumes independence between the potential outcomes of interest and the treatment variable, conditional on the confounding covariates. Because this assumption is untestable, we often estimate causal effects using all

the observed covariates. However, estimating CATEs with moderate or high dimensional covariates is challenging. For example, in Section 1 we illustrate the treatment assignment process based on one observed covariate, health score. Here, the assignment process creates a discrepancy between the treated and control populations. That is to say, we rarely observe healthy individuals who receive the treatment, and unhealthy individuals who do not receive the treatment. Then the CATE (i.e., the treatment effect conditional on the health score) becomes difficult to estimate for these individuals. The main reason for this is due to work with finite samples: the probability of observing two comparable individuals in a dataset decreases as the covariates dimension increases. However, many covariates in the high-dimensional space are often generated by some common and low-dimensional (latent) variables. Constructing an accurate CATE estimator is then easier on the low-dimensional latents than on high-dimensional observed covariates.

Contributions. In this paper, we explore the assumption that the CATE is a function with an intrinsic dimension lower than the observed covariates. We propose a representation learning method based on a partially randomized energy-based model (EBM) to embed the covariates into a low-dimensional space before estimating CATEs. This preprocessing step can be used alongside any regression model and learner to reduce their dimensionality curse in CATE estimation. We prove that the representation in the partially randomized EBM is *partially identifiable up to some universal constant* for any value of the covariates. Further, the EBM still has *universal approximation capability* for estimating any continuous covariates distribution, which avoids excessive information loss from model misspecification. To our best knowledge, identifying representations in deep learning models exactly is still infeasible. Existing theory settles on achieving weaker versions of identifiability with the help of some auxiliary information, e.g., time steps and class labels. Auxiliary information does not exist in most observational datasets. We prove that by optimizing the partially randomized EBM with a noise contrastive loss function and a sample splitting strategy, the representations can converge consistently with increasing sample sizes.

Experiments on multiple datasets complement our theoretical results. We empirically validate the convergence of the representations with increasing sample sizes. We also show that estimating CATEs based on our representations achieve better performance than directly on the covariates or the representations obtained via a variety of benchmark dimensionality reduction methods.[1]

---

[1]Code for our method is provided at: `https://github.com/vanderschaarlab/mlforhealthlabpub`.

2 Setup

Following [Neyman, 1923] and [Rubin, 1974], we use the potential outcomes framework to define causal effects. Consider an observational dataset $\mathcal{D} = \{O_i = (X_i, A_i, Y_i) : i \in [n]\}$, where $[n] = \{1, \ldots, n\}$. Each individual i is described by a set of covariates $X_i \in \mathcal{X} \subseteq \mathbb{R}^d$, a binary treatment variable $A_i \in \mathcal{A} = \{0, 1\}$ and an observed outcome $Y_i \in \mathcal{Y} \subseteq \mathbb{R}$. We assume that the samples in $\mathcal{D}$ are n i.i.d copies of the random variable

$$O = (X, A, Y) \sim \mathbb{P}(O) = \mathbb{P}(Y \mid A, X)\mathbb{P}(A \mid X)\mathbb{P}(X).$$

We assume every individual i has two potential outcomes, the control outcome $Y_i(0)$ and the treated outcome $Y_i(1)$. The treatment assignment depends on the individuals' covariates, i.e., $A_i \not\perp\!\!\!\perp X_i$. This dependence is quantified via the conditional distribution $\pi(X_i) = \mathbb{P}(A_i = 1|X_i)$, also termed as the propensity score in the literature. We make the standard assumptions for causal inference on observational data.

Assumption 2.1 (SUTVA, Ignorability and Positivity)**.** *For any* $i \in [N]$ *and* $a \in \{0, 1\}$, $Y_i = Y_i(a)$ *if* $A_i = a$. *For any* $i \in [N]$, *the distribution of* $X_i, A_i, Y_i(0)$ *and* $Y_i(1)$ *satisfies strong ignorability:* $Y_i(0), Y_i(1) \perp\!\!\!\perp A_i|X_i$ *and positivity:* $\exists \ \delta \in (0, 1)$ *s.t.* $\delta < \mathbb{P}(A_i = a|X_i = x) < 1 - \delta, \forall x \in \mathcal{X}$ *and* $a \in \{0, 1\}$.

We divide $\mathcal{D}$ into a control set and a treated set, $\mathcal{D}_\mathrm{c} = \{(X_i, A_i, Y_i) : A_i = 0, i \in [n]\}$ and $\mathcal{D}_\mathrm{t} = \{(X_i, A_i, Y_i) : A_i = 1, i \in [n]\}$. We denote the sample sizes of $\mathcal{D}_\mathrm{c}$ and $\mathcal{D}_\mathrm{t}$ by $n_\mathrm{c} = |\mathcal{D}_\mathrm{c}|$ and $n_\mathrm{t} = |\mathcal{D}_\mathrm{t}|$. Under Assumption 2.1, $\mu_a(x) := \mathbb{E}\{Y(a) \mid X = x\} = \mathbb{E}\{Y \mid X = x, A = a\}$ for $a \in \{0, 1\}$. Then we can identify the conditional average treatment effect (CATE) $\tau(x)$ by

$$\tau(x) = \mathbb{E}[Y(1) - Y(0)|X = x] = \mu_1(x) - \mu_0(x). \quad (1)$$

In nonparametric regression, the dimension and smoothness of the data generating function determine the expected squared error of a regression model [Stone, 1980]. The error of the used regression model determines the error of a CATE learner.

Definition 2.2 (Hölder ball)**.** The Hölder ball $\mathcal{H}_d(s)$ is the set of functions $f : \mathbb{R}^d \to \mathbb{R}$ supported on $\mathcal{X} \subseteq \mathbb{R}^d$ with their partial derivatives satisfying that

$$\left| \frac{\partial^m f}{\partial^{m_1} \ldots \partial^{m_d}}(x) - \frac{\partial^m f}{\partial^{m_1} \ldots \partial^{m_d}}(x') \right| \lesssim \|x - x'\|_2^{s - \lfloor s \rfloor},$$

$\forall x, x' \in \mathcal{X}$ and $m = (m_1, \cdots, m_d)$ s.t. $\sum_{j=1}^{d} m_j = \lfloor s \rfloor$.

The notation $a \lesssim b$ denotes the relation $a \leq Cb$ for some universal constant C. Essentially, $\mathcal{H}_d(s)$ is the class of smooth functions that are close to their

$\lfloor s \rfloor$-order Taylor approximations. We assume μ_0, μ_1, π and τ are s-smooth functions in the Hölder balls $\mathcal{H}_d(s)$ for some non-negative smoothness parameter $s = \alpha_0, \alpha_1, \beta, \gamma$, respectively.

The identification formula (1) motivates a common estimation strategy called a "T-learner", where "T" refers to "Two" regression models. A T-learner estimates μ_0 and μ_1 by fitting two separate regression models, $\hat{\mu}_0$ and $\hat{\mu}_1$, on $\mathcal{D}_c$ and $\mathcal{D}_t$, respectively. It estimates the CATE as the difference $\hat{\tau}(\cdot) = \hat{\mu}_1(\cdot) - \hat{\mu}_0(\cdot)$. Suppose the expected squared error of $\hat{\mu}_0$ and $\hat{\mu}_1$ are $n_c^{-\frac{2\alpha_0}{2\alpha_0+d}}$ and $n_t^{-\frac{2\alpha_1}{2\alpha_1+d}}$, respectively. A T-learner's expected squared error $\mathbb{E}\left\{[\hat{\tau}(X) - \tau(X)]^2\right\}$ is $O(n_c^{-\frac{2\alpha_0}{2\alpha_0+d}} + n_t^{-\frac{2\alpha_1}{2\alpha_1+d}})$.

There are other advanced learners based on different identification formulas, e.g., X-learner [Künzel et al., 2019], R-learner [Nie and Wager, 2021] and DR-learner [Kennedy, 2020]. For example, the identification formula of a DR-learner is based on the uncentered first-order influence function $\phi(x)$ of the ATE,

$$\phi(x) = \frac{A}{\pi(x)}\left[Y - \mu_1(x)\right] + \mu_1(x)$$
$$- \frac{1-A}{1-\pi(x)}\left[Y - \mu_0(x)\right] - \mu_0(x). \quad (2)$$

Splitting $\mathcal{D}$ into three subsets $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_3$, a DR-learner estimates the CATE as follows: estimate μ_0 and μ_1 on $\mathcal{D}_1$, estimate π on $\mathcal{D}_2$, then estimate the CATE $\tau(x)$ by regressing $\hat{\phi}(X)$ onto X in $\mathcal{D}_3$, where $\hat{\phi}(X)$ is generated by plugging the estimators $\hat{\mu}_0, \hat{\mu}_1$ and $\hat{\pi}$ into the expression (2). Kennedy [2020, Theorem 2] shows that a DR-learner's expected squared error is

$$O(\tilde{n}_2^{-\frac{2\beta}{2\beta+d}}\left(\tilde{n}_{1,c}^{-\frac{2\alpha_0}{2\alpha_0+d}} + \tilde{n}_{1,t}^{-\frac{2\alpha_1}{2\alpha_1+d}}\right) + \tilde{n}_3^{-\frac{2\gamma}{2\gamma+d}}),$$

where $\tilde{n}_m = |\mathcal{D}_m|, m = 1, 2, 3$, $\tilde{n}_{1,c}$ and $\tilde{n}_{1,t}$ are the number of control and treated individuals in $\mathcal{D}_1$. The efficiency loss from sample splitting can be remedied by cross-fitting [Nie and Wager, 2021; Chernozhukov et al., 2018]. DR-learner can improve CATE estimation by leveraging the smoothness of τ. But for an accurate CATE estimator to exist in finite samples, the requirement on the smoothness parameters ($\alpha_0, \alpha_1, \beta$ and γ) is restrictive if the amount of dimensions d is large, regardless of which learner we use. In this paper, we focus on improving CATE learners by reducing d, which is plausible under the following assumption.

Assumption 2.3. *For some positive integer $d^* < d$, there exists some variable $U \in \mathcal{U} \subseteq \mathbb{R}^{d^*}$ that generates the variables X, A and Y, i.e., the data distribution $\mathbb{P}(O)$ has a density function $p_O(o)$ which satisfies that for any $o \in \mathcal{O} = \mathcal{X} \times \mathcal{A} \times \mathcal{Y}$, $p_O(o)$ is given by*

$$\int_{\mathcal{U}} p_{Y|A,U}(y \mid a, u) p_{A|U}(a \mid u) p_{X|U}(x \mid u) p_U(u) du.$$

This assumption is realistic as observational data often includes covariates that represent the same aspect of an individual. For example, an individual's health status can be represented by some collection of covariates, e.g., some disease-specific symptoms. These covariates are correlated and contain overlapping information about the individual. Under Assumption 2.3, both the outcome Y and treatment A are generated by these abstract aspects U. This hints at a potentially more sample-efficient estimation strategy: first learning these aspects as a low-dimensional representation of the covariates, then fitting $\hat{\mu}_0, \hat{\mu}_1$ (and $\hat{\pi}$) on the low dimensional representation to estimate the CATE.

Why CATE as an application? The limitation of representation learning is that the representation itself is non-smooth and takes many samples to learn. In supervised learning on a fully labelled dataset, there is no obvious advantage of learning the representations first over learning the label directly. By contrast, observational datasets for CATE estimation are often imbalanced ($n_t \ll n_c$) so that the term $n_t^{-\frac{2\alpha_1}{2\alpha_1+d}}$ in a T-learner's expected squared error is very large. Leveraging the smoothness of τ is not very effective in the presence of high-dimensional covariates. Rather than directly using all the covariates to construct $\hat{\mu}_0, \hat{\mu}_1$ (and $\hat{\pi}$), we employ representation learning. In particular, because the representation can be learnt using *all the samples*, i.e., $n_t + n_c$ samples from both the treated and control group. Then by using the low dimensional representation to estimate CATEs, the learners will potentially have smaller expected errors.

In the next section, we present a deep neural network as our representation learning model, which allows us to keep the nonparametric merit of the regression models built on top of the representation. Under Assumption 2.1, we consider the representation learning model, concatenated together with the outcome (propensity score) model, as an outcome (propensity score) model *based on the observed covariates*. The consistency of the resulting CATE estimator thus depends on the consistency of *both* models. This requires the representation to be identifiable, which is so far still impossible to achieve exactly for overparameterized neural networks. In our next section, we provide an approximate solution to this problem, sufficient for CATE estimation.

3 Model

A parameter is identifiable in a class of statistical models if every model describing the same distribution, has the same value of the parameter. If models with different parameter-values give the same distribution, i.e., generate the same observed data in the large data limit, we can no longer find the true model from the

data even if the sample size is large [Lewbel, 2019].

Generally, identifiability is often achieved by introducing some constraint on the model class, as is also the case here. We will construct partially identifiable representations in a class of partially randomized energy-based models (EBMs). By partially identifiable, we mean if two models give the same distribution, then their representations are only different by some universal constant. A partially randomized EBM is constructed in the two steps detailed below.

Step 1. Suppose we want to learn a k-dimensional representation of the covariates $(k < d)$[2]. We let $f_\theta : \mathcal{X} \to \mathbb{R}^k$ be a neural network that generates the k-dimensional data representation. We define k standard EBMs [LeCun et al., 2006] on $\mathcal{X}$ with a *shared* representation f_θ:

$$p_{\theta,j}(x) = Z_{\theta,j}^{-1} \exp\left[-\beta_j^\top f_\theta(x)\right], \ \forall j \in [k], \quad (3)$$

where $Z_{\theta,j} = \int_{\mathcal{X}} \exp\left[-\beta_j^\top f_\theta(x)\right] dx$. The number of standard EBMs is the same as the size of the representation for a purpose. Roughly speaking, we want to create k equations to determine a k-dimensional representation (for details, we refer to Proposition 3.3 and its proof in Appendix B.2). The EBM (3) can be written as an exponential family distribution, $p_{\theta,j}(x) = h(x)\exp\left[\lambda_j^\top f_\theta(x) - \psi_{\theta,j}\right]$, where $h(x) = 1$ and $\lambda_j = -\beta_j$, and $\psi_{\theta,j} = \log\left(\int_{\mathcal{X}} h(x)\exp\left[\lambda_j^\top f_\theta(x)\right] dx\right)$.

Proposition 3.1. *For every $j \in [k]$, f_θ is a minimum and sufficient statistic in model (3).*

Step 2. Let $\mathcal{P} = \{p_{\theta,j} \mid \beta_j \in \mathbb{R}^k, \theta \in \Theta\}$ denote the space of standard EBMs, and $\mathcal{P}(\beta_j)$ denote the subset of $\mathcal{P}$ with β_j fixed. Let $B = (\beta_1, \ldots, \beta_k)$ be the $k \times k$ matrix whose j-th column is β_j. A partially randomized EBM is given by multiple standard EBMs with a shared representation f_θ and a fixed random orthogonal matrix B, i.e., θ is the only learnable parameter.

Definition 3.2 (Partially Randomized EBM). A partially randomized EBM is given by

$$p_\theta = (p_{\theta,j} : j \in [k]) \in \bigtimes_{j=1}^{k} \mathcal{P}(\beta_j), \quad (4)$$

where $B = (\beta_1, \ldots, \beta_k)$ is a $k \times k$ random orthogonal matrix s.t. $BB^\top = I_{k \times k}$.

One easy approach to construct B is to first generate a random matrix $B_0 \in \mathbb{R}^{k \times k}$, where each entry is drawn independently from a standard normal distribution,

---

[2]The errors of CATE learners depend on the performance of the outcome and propensity score models $\hat{\mu}_0$, $\hat{\mu}_1$ and $\hat{\pi}$. Because some CATE learners do not use a propensity score model, the dimension k is tuned as a hyper-parameter via cross-validation on the observed outcomes in this paper.

and then taking B as the matrix of eigenvectors of B_0. The partially randomized EBM satisfies the partial identifiability defined as follows.

Proposition 3.3. *For any $k \times k$ random orthogonal matrix $B = (\beta_1, \ldots, \beta_k)$ and $p_{\theta,j}, p_{\tilde{\theta},j} \in \mathcal{P}(\beta_j)$ such that $p_{\theta,j}(\cdot) = p_{\tilde{\theta},j}(\cdot), \forall j \in [k]$, we have*

$$f_\theta(\cdot) - f_{\tilde{\theta}}(\cdot) = C \ \ \text{for some constant vector } C. \quad (5)$$

Perhaps surprisingly, the randomization strategy above does not overly decrease the model complexity. Proposition 3.4 verifies the universal approximation capability of the partially randomized EBM. The proof is attained by showing that $\mathcal{P}(\beta_j)$ satisfies the conditions in the Stone-Weierstrass approximation theorem. The proofs of all the propositions can be found in Appendix B.

Proposition 3.4. *For any continuous density function $p_X : \mathcal{X} \to \mathbb{R}^+$, $k \times k$ random orthogonal matrix B, and $\epsilon > 0$, there exists $p_{\theta,j} \in \mathcal{P}(\beta_j)$ such that $\sup_{x \in \mathcal{X}} |p_X(x) - p_{\theta,j}(x)| \le \epsilon$ for all $j \in [k]$.*

Next, we will introduce a training strategy for our partially randomized EBM, which will enable the learnt representation model to converge to a limiting set in which any functions are only different by some constants C like f_θ and $f_{\tilde{\theta}}$ in (5). This will enable the follow-up CATE estimates to converge consistently because the regression models $\hat{\mu}_0, \hat{\mu}_1$ (and $\hat{\pi}$) are indifferent to conditioning on a random variable, or the same random variable plus some constant vector. Furthermore, by standardizing the learnt representations, we can fix their mean to 0 and their variance to 1 in any sample size. Given that the representations have mean 0 and variance 1, by partial identifiability and consistency, the representations obtained from different runs of the experiments will have a correlation close to 1 at each dimension in large samples, as will be demonstrated experimentally in Section 6.2.

4 Method

Fitting energy-based models (EBMs) by maximum likelihood estimation (MLE) is often infeasible because the partition function $(Z_{\theta,j})$ is intractable. Noise Contrastive Estimation (NCE) proposed by [Gutmann and Hyvärinen, 2010, 2012] is a consistent and computationally efficient alternative. The high-level idea of NCE is to optimize an EBM by contrasting it with another noise distribution with known and easy-to-sample density. Advanced methods have been proposed to tune the noise distribution, see for example Gao et al. [2020]; Bose et al. [2018]; Ceylan and Gutmann [2018].

Here for every individual $i \in [n]$, we draw b corrupted samples $\tilde{X}_{i1}, \ldots, \tilde{X}_{ib}$ from a noise distribution

$p_{\tilde{X}|X}(\tilde{x} \mid X_i)$ defined as follows. Each $\tilde{X}_{ia}$ is generated in two steps: (1) we sample an independent binary variable with some probability for each feature of X_i, used to decide which features of X_i will be corrupted, then (2) corrupt each selected continuous feature by adding white noise drawn from a standard normal distribution, and corrupt each selected categorical feature by uniformly sampling a value from its range. A mathematical description of $p_{\tilde{X}|X}(\tilde{x} \mid X_i)$ is provided in Appendix B.6. Overall, the original and corrupted data of individual i is given by

$$\bar{X}_i = (X_i, \tilde{X}_{i1}, \ldots, \tilde{X}_{ib}) \sim p_X(x) \prod_{a=1}^{b} p_{\tilde{X}|X}(\tilde{x} \mid x).$$

We split the n individuals into k subsets $\mathcal{I}_j, j \in [k]$, to train each of of k models $p_{\theta,j}(x)$ in the partially randomized EBM as in (4). Suppose we randomly permute the columns of $\bar{X}_i$ and let $V_i = (V_{ia} : a \in [b+1])$ be the permuted $\bar{X}_i$. Then each column of V_i has equal probability $(b+1)^{-1}$ for being the original sample X_i. We derive the predictive probability of $V_{ia} = X_i$ from the posterior distribution,

$$q_{\theta,j}(a \mid V_i) = \frac{(b+1)^{-1} p_{\theta,j}(V_{ia}) \tilde{p}_{-a}(V_i)}{\sum_{c=1}^{b+1} (b+1)^{-1} p_{\theta,j}(V_{ic}) \tilde{p}_{-c}(V_i)}, \quad (6)$$

where $\tilde{p}_{-a}(V_i) = \prod_{a \in [b+1]:a' \neq a} \tilde{p}_{\tilde{X}|X}(V_{ia'} \mid V_{ia})$. It is noteworthy that the intractable partition function $Z_{\theta,j}$ in $p_{\theta,j}$ (in (3)) cancels out in the expression of $q_{\theta,j}(a \mid V_i)$. Let $W_i \in \{0,1\}^{b+1}$ indicate which column of V_i is X_i. We can think of $\{(V_i, W_i) : i :\in \mathcal{I}_j\}$ as a set of labeled "images" and optimize the probability $q_{\theta,j}(a \mid V_i)$ to predict W_i. Let $n_j = |\mathcal{I}_j|$. Our objective function is the negative cross-entropy[3], $\mathcal{L}_n(\theta) = k^{-1} \sum_{j=1}^{k} \mathcal{L}_{n,j}(\theta)$, where $\mathcal{L}_{n,j}(\theta)$ is given by

$$\begin{aligned} \mathcal{L}_{n,j}(\theta) &= n_j^{-1} \sum_{i \in \mathcal{I}_j} \sum_{a=1}^{b+1} W_{ia} \log q_{\theta,j}(a \mid V_i) \\ &= n_j^{-1} \sum_{i \in \mathcal{I}_j} \log q_{\theta,j}(1 \mid \bar{X}_i). \end{aligned} \quad (7)$$

The representation model f_θ is trained on all the samples, even though we split the samples across the models $p_{\theta,j}, j \in [k]$, in our partially randomized EBM.

The training strategy here follows the same principle as the other representation learning methods, e.g., [Vincent et al., 2010; Vincent, 2011]: assume the covariates X_i live in some d^*-dimensional manifold ($d^* < d$). If $q_{\theta,j}(a \mid V_i)$ is predictive of W_i, i.e., can distinguish

---

[3]This is essentially the ranking objective in [Józefowicz et al., 2016; Ma and Collins, 2018] with a different noise distribution. We reformulate the training strategy as a more intuitive multiclass classification task.

any true sample $X_i \sim p_X(x)$ from its noisy proxies $\tilde{X}_{ia} \sim \tilde{p}_{\tilde{X}|X}(\tilde{x} \mid X_i)$, we have $p_{\theta,j}(x) \approx p_X(x)$ in (6). This implies that the low-dimensional representation given by $f_\theta(x)$ is informative of the true covariates X_i; the representation is also predictive of the outcome and treatment because they are generated by the covariates.

Proposition 4.1 below shows that by our training strategy, the learnt parameter $\hat{\theta}_n$ will converge to a set of limits Θ_0 s.t. $p_{\theta_0,j}(x) = p_X(x)$ for any $x \in \mathcal{X}$ and $\theta_0 \in \Theta_0$. Then by (5) in Proposition 3.3, no matter which $\theta_0 \in \Theta_0$ that $\hat{\theta}_n$ converges to, the limit of $f_{\hat{\theta}_n}(x)$ will be only different by some universal constant.

Proposition 4.1. *Suppose that the covariates space $\mathcal{X}$ is a compact subset of $\mathbb{R}^d$, $f_\theta(x)$ has a compact parameter space Θ, and $f_\theta(x)$ is continuous with respect to its parameter θ for any $x \in \mathcal{X}$. For any $k \times k$ random orthogonal matrix B, under Proposition 3.4, we assume for any continuous density function $p_X(x)$ defined on $\mathcal{X}$, there exists a countable subset $\Theta_0 \subset \Theta$ s.t. $p_{\theta_0,j}(x) = p_X(x)$ for any $x \in \mathcal{X}$ and $\theta_0 \in \Theta_0$. For any number of noise samples b and $\hat{\theta}_n \in \arg\max_{\theta \in \Theta} \mathcal{L}_n(\theta)$, we have $\lim_{n \to \infty} \hat{\theta}_n \in \Theta_0$ with probability 1.*

Essentially, both MLE and NCE are special cases of M-estimators in statistics [Van der Vaart, 2000]. The proposition is proven by showing that $\mathcal{L}_{\infty,j}(\theta)$ is maximized by $q_{\theta,j}(a \mid V_i)$ with $p_{\theta,j}(x) = p_X(x)$, and the standard conditions for consistent M-estimators hold for $\hat{\theta}_n$ under a weaker identifiability assumption; see Appendix B.4 for more details.

5 Related works

Here we provide related works on three different areas.

Identifiability theory. Khemakhem et al. [2020b] propose two definitions of identifiability for EBMs; weak and strong identifiability (in their Definitions 1 and 2). Their EBM is more complex than ours with β_j as a learnable parameter, while their objective is to identify both β_j and $f_\theta(x)$. This is unnecessary for the application in our paper. Arguably, the partial identifiability defined in our paper is stronger than both of their definitions. In their strong identifiability, under some assumptions, each dimension of f_θ is identifiable up to be multiplied by and plus some constants, and each dimension of f_θ can be permuted in any order. They also require a specific network architecture for f_θ. In our work, we use a *simpler* partially randomized EBM to achieve a *stricter* version of identifiability, without sacrificing the approximation capability of the EBM or restricting the architecture of f_θ.

The works on nonlinear ICA and its generalization [Hyvarinen and Morioka, 2016; Hyvarinen et al., 2019;

Khemakhem et al., 2020a; Mita et al., 2021] propose the idea of using contrastive learning for identifiable feature extraction when some auxiliary information (e.g., time steps) about the features is available. We use sample splitting and a noise contrastive loss function for training the partially randomized EBM, assuming *no auxiliary information* is provided in the observational data. Monti et al. [2020] and Wu and Fukumizu [2020] propose non-linear ICA based methods for causal inference on structural causal models [Pearl, 2009]. The setup and problems studied in their papers are different from our method which is developed within the potential outcomes framework.

Representation learning. Representation learning is recently applied to balance or match the covariate distribution between the treated and control group in observational data, by minimizing the distributional distance between the group [Shalit et al., 2017], preserving local similarity [Yao et al., 2018], minimizing counterfactual variance [Zhang et al., 2020] and adversarial training [Kallus, 2020]. We note that supervised dimensionality reduction in a deep learning model is not reliable because the model can easily overfit the limited outcome data without finding an informative representation of the covariates. Our proposed method works more generally as a preprocessing step to reduce the dimentionality curse for any regression model, including these deep learning models which balance the distribution in one of their hidden layers.

In statistics, sufficient dimensionality reduction (SDR) [Lee et al., 2013; Cook, 2009; Li, 1991; Adragni and Cook, 2009] has been used in the models for estimating ATE and CATE [Huang and Yang, 2022; Luo et al., 2019; Cheng et al., 2020; Ma et al., 2019; Ghosh et al., 2018]. If the subspace spanned by the columns of a $d \times k$ matrix θ with $k \leq d$ satisfies that $Y \perp\!\!\!\perp X \mid \theta^\top X$, we call this subspace a SDR subspace. The idea of SDR is to project the covariates X onto this subspace before feeding it into a parametric or nonparametric regression model to estimate Y. To achieve the desired conditional independence, θ is jointly learnt with the regression model. This is not straightforward for some of the ML models. e.g., decision tree. Kallus et al. [2018] proposes a matrix factorization based method for preprocessing noisy and missing covariates. In contrast with these methods, our method performs nonlinear dimensionality reduction of the covariates, which is more general for the data living in some low-dimensional manifold, including linear subspace. Nabi and Shpitser [2020] and Berrevoets et al. [2020] propose methods to deal with high-dimensional treatment variables, which is not the problem considered in our paper.

Covariates selection. When there are irrelevant covariates in a dataset, data analysis should start with a covariates selection method, e.g., [De Luna et al., 2011; Shortreed and Ertefaie, 2017; Greenewald et al., 2021]. However, covariates selection methods often have no guarantee to find the correct adjustment set for causal inference in finite samples. Furthermore, (selected) covariates are correlated, especially when we allow more covariates to be selected in order to satisfy the strong ignorability assumption. On the basis of this, our representation learning method can be applied to further reduce the dimensionality of the correlated covariates and improve the accuracy of CATE estimation. In general, covariates selection and our method are applied in different stages and complement each other in the data analysis process.

6 Experiments

We make two claims in our paper: (1) using our method as a preprocessing step increases the performance of CATE learners; (2) the representation in our model (4) is partially identifiable so that the learnt representations and downstream CATE estimates are consistent. We test these two claims in the following subsections. Throughout our experiments, we use four different CATE learners: X-Learner, DR-learner, T-Learner, and R-learner [Microsoft Research, 2019]. We provide more details of our experiments (e.g., on learners and hyperparameters) in Appendix A.

6.1 CATE estimation

Our main contribution is a way to increase performance for *any* learner. Specifically, in high dimensions and small sample sizes. We evaluate learners' performance using *precision of estimating heterogeneous effects* (PEHE) introduced in Hill [2011] and now standard in CATE estimation. PEHE is essentially the expected risk $\mathbb{E}\left\{[\hat{\tau}(X) - \tau(X)]^2\right\}$ we define in Section 2. Because any individual's treated and control outcomes are never observed jointly, CATEs are unobserved in any real-world data. The literature thus relies on (semi-)synthetic data to evaluate CATE learners.

In our synthetic setup, the generating process of the observed variables $O = (X, A, Y)$ starts by sampling a latent variable $U \sim \mathcal{N}(0, I_{5\times5})$. Then we generate a set of covariates $X = \mathcal{N}(g(U), I_{d\times d})$, two potential outcomes, $\mu_0(U)$ and $\mu_1(U)$ and a treatment assignment $A \sim \text{Ber}[\pi(U)]$. The observed outcome is given by $Y = \mathcal{N}(A\mu_0(U) + (1-A)\mu_1(U), 1)$. The CATE is given by $\tau(U) = \mu_1(U) - \mu_0(U)$. The function g is a deep ReLU network; μ_0 and μ_1 are one-layer neural networks, with an exp-function on their output layers; π is a one-layer network with a sigmoid-function on its output layer. By generating i.i.d samples from this process, we create a training set (with size n specified

Yao Zhang*, Jeroen Berrevoets*, Mihaela van der Schaar

Table 1: **Results on synthetic data and semi-synthetic data (Twins).** Each row reports the average PEHE (lower is better) over ten runs for each CATE learner (standard deviation in scriptsize): both *with* representations (indicated as "✔"), and *without* representation (indicated as "✗"). For each run, we learn a new representation. In the above two blocks, we vary sample sizes and dimensions using our synthetic setup, and in the bottom block we vary the sample size for the Twins-dataset. Using our EBM yields superior testing performance for a range of CATE learners (indicated in bold). In green , we emphasize the best results per row, each time *with* EBM. While there may be duplicate values, we highlight only those that are best beyond the rounding applied here.

Methods	X-Learner		DR-Learner		T-Learner		R-Learner	
EBM	✗	✔	✗	✔	✗	✔	✗	✔
d n	*Synth. data with increasing sample size and increasing dimensions*							
50 100	2.309 $\pm$.00	**1.994** $\pm$.02	4.594 $\pm$.56	**2.017** $\pm$.04	2.441 $\pm$.00	**1.993** $\pm$.01	3.194 $\pm$.26	**1.982** $\pm$.04
100 250	2.779 $\pm$.00	**2.018** $\pm$.01	4.056 $\pm$.32	**2.154** $\pm$.39	2.838 $\pm$.00	**2.019** $\pm$.01	3.702 $\pm$.23	**2.018** $\pm$.01
150 500	2.618 $\pm$.00	**2.000** $\pm$.01	3.030 $\pm$.12	**2.001** $\pm$.01	2.641 $\pm$.00	**2.000** $\pm$.01	2.877 $\pm$.08	**2.000** $\pm$.01
200 1k	2.185 $\pm$.00	**1.940** $\pm$.01	2.283 $\pm$.02	**1.941** $\pm$.01	2.189 $\pm$.00	**1.939** $\pm$.01	2.271 $\pm$.01	**1.940** $\pm$.01
250 1.5k	2.267 $\pm$.00	**1.949** $\pm$.02	2.427 $\pm$.01	**1.976** $\pm$.00	2.271 $\pm$.00	**1.948** $\pm$.01	2.436 $\pm$.02	**1.949** $\pm$.02
n	*Synth. data with increasing sample size and dimensions fixed at $d = 100$*							
100	2.134 $\pm$.00	**1.927** $\pm$.01	24.61 $\pm$9.9	**2.096** $\pm$.09	2.279 $\pm$.00	**1.929** $\pm$.01	3.192 $\pm$.13	**1.925** $\pm$.01
250	2.779 $\pm$.00	**2.018** $\pm$.01	4.056 $\pm$.32	**2.154** $\pm$.39	2.838 $\pm$.00	**2.019** $\pm$.01	3.702 $\pm$.23	**2.018** $\pm$.01
500	2.155 $\pm$.00	**2.056** $\pm$.02	2.334 $\pm$.07	**2.273** $\pm$.67	2.166 $\pm$.00	**2.053** $\pm$.02	2.271 $\pm$.05	**2.056** $\pm$.02
1k	2.059 $\pm$.00	**1.964** $\pm$.02	2.105 $\pm$.01	**2.016** $\pm$.16	2.061 $\pm$.00	**1.964** $\pm$.02	2.086 $\pm$.01	**1.965** $\pm$.02
1.5k	2.013 $\pm$.00	**1.998** $\pm$.02	2.043 $\pm$.01	**1.998** $\pm$.02	2.014 $\pm$.00	**1.998** $\pm$.02	2.024 $\pm$.01	**1.991** $\pm$.02
n	*Twins ($d = 48$) with increasing sample size*							
500	0.214 $\pm$.00	**0.144** $\pm$.00	0.236 $\pm$.04	**0.182** $\pm$.05	0.221 $\pm$.00	**0.145** $\pm$.00	0.222 $\pm$.02	**0.145** $\pm$.00
1k	0.294 $\pm$.00	**0.162** $\pm$.00	0.348 $\pm$.12	**0.173** $\pm$.03	0.301 $\pm$.00	**0.162** $\pm$.01	0.532 $\pm$.11	**0.161** $\pm$.00
1.5k	0.165 $\pm$.00	**0.154** $\pm$.00	0.189 $\pm$.06	**0.159** $\pm$.01	0.165 $\pm$.00	**0.154** $\pm$.00	0.172 $\pm$.01	**0.154** $\pm$.00
2k	0.167 $\pm$.00	**0.156** $\pm$.00	0.197 $\pm$.03	**0.159** $\pm$.00	0.167 $\pm$.00	**0.156** $\pm$.00	0.222 $\pm$.05	**0.157** $\pm$.00
2.5k	0.297 $\pm$.00	**0.153** $\pm$.00	0.390 $\pm$.19	**0.156** $\pm$.00	0.297 $\pm$.00	**0.153** $\pm$.00	0.358 $\pm$.22	**0.153** $\pm$.00

in Table 1) and a large testing set with 20k samples.

Given a training set, we first use it to optimize our partially randomized EBM. Then we preprocess it and apply various CATE learners on these lower-dimensional representations. As a comparison, we also apply the same CATE learners on the original covariates.

Lower PEHE across CATE learners. Table 1 shows that our method greatly benefits a broad spectrum of CATE learners on the synthetic dataset and semi-synthetic dataset Twins [Almond et al., 2005] with real covariates, especially in small sample sizes. While the gain of using our method diminishes somewhat in larger sample sizes, it is still significant. More importantly, we observe that with our EBM, the performance gaps between different learners shrink significantly. Specifically, R-learner with EBM has the best performance on average over the table while it performs poorly in small samples without EBM. Overall, our experimental results align with our theoretical discussion in Section 2: by reducing the dimensionality d to a smaller number, the learners will have lower expected errors, i.e., lower PEHEs and smaller performance gaps.

Lower PEHE than benchmark dimensionality reduction methods. Based on our previous experiment, a logical next question to ask is whether other dimensionality reduction methods may also help. We compare our EBM method to various linear and nonlinear dimensionality reduction methods in prepossessing the real covariates of the Twins dataset. Specifically, we compare against: Principal Components Analysis (PCA), Feature Agglomeration (FA), Spectral Embedding (SE), Isomap, KernelPCA with an RBF kernel, and an Autoencoder (AE). Table 2 shows that our EBM method outperforms all the benchmarks significantly over different sample sizes.

To further validate our proposed method, we repeat the same experiment using **additional regression models and data**, and report consistent results to those we present in this section, in Appendix A. Overall, we do not found **sample splitting** increase the variance of our method across all our experiments. As we explained below eq. (7), the representation model f_θ is trained with all the samples in our objective function.

Table 2: **Results using different dimensionality reduction methods.** Using an R-learner, we report the PEHE of our EBM and other benchmark methods over 10 runs (standard deviation in scriptsize): PCA, Feature Agglomeration (FA), Spectral Embedding (SE), Isomap, and KernelPCA (K-PCA) and Autoencoder (AE).

Methods	**PCA**	**FA**	**SE**	**Isomap**	**K-PCA**	**AE**	**EBM**
n	*Twins* ($d = 48$) *with increasing sample size*						
500	1.092 ±.11	1.758 ±1.1	1.011 ±.00	1.006 ±.00	1.015 ±.00	0.580 ±.03	**0.145** ±.00
1k	1.015 ±.00	0.963 ±.00	1.010 ±.00	1.004 ±.00	1.010 ±.00	0.549 ±.04	**0.161** ±.00
1.5k	1.014 ±.00	0.965 ±.00	1.005 ±.00	1.006 ±.00	1.012 ±.00	0.546 ±.04	**0.154** ±.00
2k	1.013 ±.00	0.957 ±.00	1.009 ±.00	1.007 ±.00	1.013 ±.00	0.579 ±.03	**0.157** ±.00
2.5k	1.007 ±.00	0.951 ±.00	1.002 ±.00	1.006 ±.00	1.006 ±.00	0.542 ±.04	**0.153** ±.00

6.2 Partial identifiability of representations

In this section we empirically validate that our method produces identifiable representations. Having an identifiable method is important for later inspection of the representations, but also to produce consistent CATE learners. Both of which are important in practice.
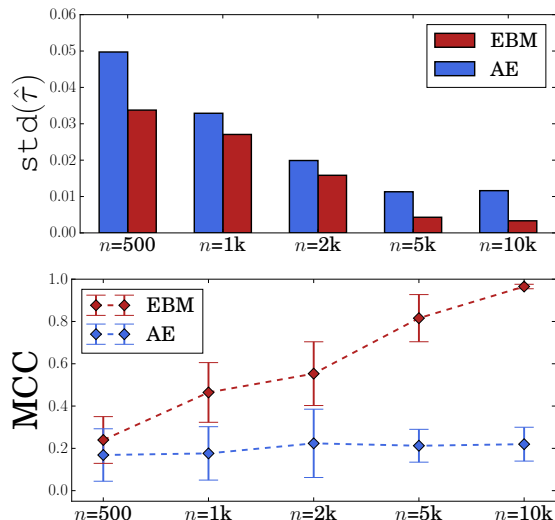

Figure 2: **Results on identifiability.** *Above*— For each model (an autoencoder (AE), and our model (EBM)) we learn ten distinct representations. We then fit an R-Learner on each representation, and calculate the standard deviation of their CATE estimates. Our method has lower standard errors compared to AE. *Below*— We report the mean correlation coefficient (MCC) between the representations on the Twins data (higher is better). Our EBM becomes more consistent with larger samples (error bars indicate standard deviation on MCC), and even tends to 1 in large samples.

Converging CATE estimates. The first panel in Figure 2 reports the standard deviation of the CATE-estimates, by an R-learner when fitted on the representations of: an autoencoder (AE) and our method (EBM). The representations have the same amount of dimensions ($k = 5$). Figure 2 shows that our model decreases the standard deviation with increasing sample size—this is important, as many applications require estimates to be consistent.

Converging representations. As discussed at the end of Section 4, the learnt representations after standardization should correlate as sample size increases. We train our EBM ten times using distinct random initializations, while keeping the random matrix fixed across runs. We subsequently compute the mean correlation coefficient (MCC) between the representations of the test-set from different runs. The MCC is computed by averaging the correlation between *each dimension* in the representations of 20k samples from the test set. Note that the latter is a strict definition as it requires the representation to be consistent for each *individual* dimension.[4] Reported in the second panel, we see that our EBM's MCC grows as the sample size increases, leaving the (unidentifiable) AE behind, indicating that our EBM is identifiable, further confirming our theory.

7 Conclusions

We propose a partially randomized EBM with universal approximation capability to learn a partially identifiable low-dimensional representation of moderate or high-dimensional covariates in CATE estimation. We show theoretically and empirically that by training our EBM with a noise contrastive loss function and a sample splitting strategy, our representations converge to a set of limits differing only by some constants. This enables downstream learners to achieve consistent CATE estimates. Experiments on multiple datasets with various dimensions and sample sizes verify our theories and demonstrate a significant performance increase when using our method for CATE estimation.

Our work opens a few new directions for future re-

---

[4]Previous work [Khemakhem et al., 2020b] tests identifiability using the MCC maximized by canonical-correlation analysis (CCA). Here we compute the exact correlation to test our stronger version of identifiability.

Yao Zhang[*], Jeroen Berrevoets[*], Mihaela van der Schaar

search. First, our method currently operates within the standard setup of observational data in causal inference, while our partial identifiability theory does not rely on the network architecture of f_θ. Extending our work to other high-dimensional settings such as time series or vision could prove useful for many real world applications. Second, as an interpretable approach within CATE estimation, matching is concerned with finding similar individuals across treatment and control groups. While effective, matching becomes harder in high-dimensions. Extending our approach to remain interpretable (e.g. by measuring each covariate's influence to each dimension in the representation) can arm matching approaches against the dimensionality curse. In essence, we believe our method could benefit a wide range of applications requiring causal inference.

8 Acknowledgements

References

Adragni, K. P. and Cook, R. D. (2009). Sufficient dimension reduction and prediction in regression. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1906):4385–4405.

Almond, D., Chay, K. Y., and Lee, D. S. (2005). The costs of low birth weight. *The Quarterly Journal of Economics*, 120(3):1031–1083.

Bartlett, P. L., Montanari, A., and Rakhlin, A. (2021). Deep learning: a statistical viewpoint. *arXiv preprint arXiv:2103.09177*.

Berrevoets, J., Jordon, J., Bica, I., Gimson, A., and van der Schaar, M. (2020). OrganITE: Optimal transplant donor organ offering using an individual treatment effect. In *Advances in Neural Information Processing Systems*, volume 33, pages 20037–20050. Curran Associates, Inc.

Biewald, L. (2020). Experiment tracking with weights and biases. Software available from wandb.com.

Bose, A., Ling, H., and Cao, Y. (2018). Adversarial contrastive estimation. In *ACL*.

Ceylan, C. and Gutmann, M. (2018). Conditional noise-contrastive estimation of unnormalised models. In *ICML*.

Cheng, D., Li, J., Liu, L., and Liu, J. (2020). Sufficient dimension reduction for average causal effect estimation. *arXiv preprint arXiv:2009.06444*.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. (2018). Double/debiased machine learning for treatment and structural parameters.

Cook, R. D. (2009). *Regression graphics: Ideas for studying regressions through graphics*, volume 482. John Wiley & Sons.

De Luna, X., Waernbaum, I., and Richardson, T. S. (2011). Covariate selection for the nonparametric estimation of an average treatment effect. *Biometrika*, 98(4):861–875.

Folland, G. B. (1999). *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons.

Gao, R., Nijkamp, E., Kingma, D. P., Xu, Z., Dai, A. M., and Wu, Y. N. (2020). Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7518–7528.

Ghosh, T., Ma, Y., and de Luna, X. (2018). Sufficient dimension reduction for feasible and robust estimation of average causal effect. *arXiv preprint arXiv:1811.01992*.

Greenewald, K., Shanmugam, K., and Katz, D. (2021). High-dimensional feature selection for sample efficient treatment effect estimation. In *International Conference on Artificial Intelligence and Statistics*, pages 2224–2232. PMLR.

Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304. JMLR Workshop and Conference Proceedings.

Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(2).

Hill, J. L. (2011). Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240.

Huang, M.-Y. and Yang, S. (2022). Robust inference of conditional average treatment effects using dimension reduction. *Statistica Sinica*, 32:1–21.

Hyvarinen, A. and Morioka, H. (2016). Unsupervised feature extraction by time-contrastive learning and nonlinear ica. *Advances in Neural Information Processing Systems*, 29:3765–3773.

Hyvarinen, A., Sasaki, H., and Turner, R. (2019). Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 859–868. PMLR.

Jennrich, R. I. (1969). Asymptotic properties of nonlinear least squares estimators. *The Annals of Mathematical Statistics*, 40(2):633–643.

Józefowicz, R., Vinyals, O., Schuster, M., Shazeer, N. M., and Wu, Y. (2016). Exploring the limits of language modeling. *ArXiv*, abs/1602.02410.

Kallus, N. (2020). Deepmatch: Balancing deep covariate representations for causal inference using adversarial training. In *International Conference on Machine Learning*, pages 5067–5077. PMLR.

Kallus, N., Mao, X., and Udell, M. (2018). Causal inference with noisy and missing covariates via matrix factorization. In *NeurIPS*.

Kennedy, E. H. (2020). Optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*.

Khemakhem, I., Kingma, D., Monti, R., and Hyvarinen, A. (2020a). Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR.

Khemakhem, I., Monti, R., Kingma, D., and Hyvarinen, A. (2020b). Ice-beem: Identifiable conditional energy-based deep models based on nonlinear ica. *Advances in Neural Information Processing Systems*, 33.

Künzel, S. R., Sekhon, J. S., Bickel, P. J., and Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. (2006). A tutorial on energy-based learning. *Predicting structured data*, 1(0).

Lee, K.-Y., Li, B., Chiaromonte, F., et al. (2013). A general theory for nonlinear sufficient dimension reduction: Formulation and estimation. *Annals of Statistics*, 41(1):221–249.

Lewbel, A. (2019). The identification zoo: Meanings of identification in econometrics. *Journal of Economic Literature*, 57(4):835–903.

Li, K.-C. (1991). Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327.

Luo, W., Wu, W., and Zhu, Y. (2019). Learning heterogeneity in causal inference using sufficient dimension reduction. *Journal of Causal Inference*, 7(1).

Ma, S., Zhu, L., Zhang, Z., Tsai, C.-L., and Carroll, R. J. (2019). A robust and efficient approach to causal inference based on sparse sufficient dimension reduction. *Annals of statistics*, 47(3):1505.

Ma, Z. and Collins, M. (2018). Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *EMNLP*.

MacDorman, M. F. and Atkinson, J. O. (1999). Infant mortality statistics from the 1997 period linked birth/infant death data set. *National vital statistics reports : from the Centers for Disease Control and Prevention, National Center for Health Statistics, National Vital Statistics System*, 47:1–23.

Microsoft Research (2019). EconML: A Python Package for ML-Based Heterogeneous Treatment Effects Estimation. https://github.com/microsoft/EconML. Version 0.11.1.

Mita, G., Filippone, M., and Michiardi, P. (2021). An identifiable double vae for disentangled representations. In *International Conference on Machine Learning*, pages 7769–7779. PMLR.

Monti, R. P., Zhang, K., and Hyvärinen, A. (2020). Causal discovery with general non-linear relationships using non-linear ica. In *Uncertainty in Artificial Intelligence*, pages 186–195. PMLR.

Nabi, R. and Shpitser, I. (2020). Semi-parametric causal sufficient dimension reduction of high dimensional treatments. *arXiv preprint arXiv:1710.06727*.

Newey, W. K. and McFadden, D. (1994). Large sample estimation and hypothesis testing. *Handbook of econometrics*, 4:2111–2245.

Neyman, J. S. (1923). On the application of probability theory to agricultural experiments. essay on principles. section 9. *Annals of Agricultural Sciences*, 10:1–51. (Translated to English and edited by D. M. Dabrowska and T. P. Speed, *Statistical Science* (1990), 5, 465–480).

Nie, X. and Wager, S. (2021). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319.

Pearl, J. (2009). *Causality*. Cambridge university press.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

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

Rosenbaum, P. R. and Rubin, D. B. (1984). Reducing bias in observational studies using subclassification on the propensity score. *Journal of the American statistical Association*, 79(387):516–524.

Rubin, D. B. (1974). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688.

Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pages 3076–3085. PMLR.

Shortreed, S. M. and Ertefaie, A. (2017). Outcome-adaptive lasso: variable selection for causal inference. *Biometrics*, 73(4):1111–1122.

Stone, C. J. (1980). Optimal rates of convergence for nonparametric estimators. *The annals of Statistics*, pages 1348–1360.

Van der Vaart, A. W. (2000). *Asymptotic statistics*, volume 3. Cambridge university press.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, 11:3371–3408.

Wooldridge, J. M. (2010). *Econometric analysis of cross section and panel data*. MIT press.

Wu, P. and Fukumizu, K. (2020). Causal mosaic: Cause-effect inference via nonlinear ica and ensemble method. In *International Conference on Artificial Intelligence and Statistics*, pages 1157–1167. PMLR.

Yao, L., Li, S., Li, Y., Huai, M., Gao, J., and Zhang, A. (2018). Representation learning for treatment effect estimation from observational data. *Advances in Neural Information Processing Systems*, 31.

Yeo, I.-K. and Johnson, R. A. (2000). A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959.

Zhang, Y., Bellot, A., and Schaar, M. (2020). Learning overlapping representations for the estimation of individualized treatment effects. In *International Conference on Artificial Intelligence and Statistics*, pages 1005–1014. PMLR.

A Additional experiments & Hyperparameters

In Appendix A.1 we compare a variety of dimensionality reduction methods to our EBM, by using each as a preprocessing step before constructing a CATE estimator. In Appendix A.2 we repeat our experiments in Table 1 using different regression models to estimate the outcomes and propensity score in the same CATE learners, and using an additional real-world dataset. For hyperparameter settings we refer to Appendix A.3, and for details on used CATE learners we refer to Appendix A.4.

A.1 Comparision to alternative dimensionality reduction methods

In Table 3 we compare our EBM to a variety of alternative dimensionality reduction methods. Specifically, we compare against: principal components analysis (PCA), Feature Agglomeration (FA), Spectral Embedding (SE), Isomap, KernelPCA with an RBF kernel, and Autoencoder (AE). From Table 3 we note that none of these methods succeeds in successfully learning informative representations, in such a way to keep downstream CATE learners accurate; our EBM outperforms the benchmarks in most cases over different datasets with various dimensions and sample sizes.

Table 3: **Results using different dimensionality reduction methods (Copy of Table 2 with additional data).** Using an R-learner, we report the PEHE of our EBM and other benchmark methods: PCA, Feature Agglomeration (FA), Spectral Embedding (SE), Isomap, and KernelPCA (K-PCA) and Autoencoder (AE).

Methods		PCA	FA	SE	Isomap	K-PCA	AE	EBM
d	n|		*Synth. data with increasing sample size and increasing dimensions*					
50	100	$2.139 \pm.00$	$2.123 \pm.00$	$2.183 \pm.00$	$2.135 \pm.00$	$2.141 \pm.00$	$2.259 \pm.02$	$\mathbf{1.982} \pm.01$
100	250	$2.238 \pm.00$	$2.239 \pm.00$	$2.151 \pm.00$	$2.231 \pm.00$	$2.236 \pm.00$	$2.055 \pm.01$	$\mathbf{2.032} \pm.01$
150	500	$2.224 \pm.00$	$2.209 \pm.00$	$2.963 \pm.00$	$2.231 \pm.00$	$2.239 \pm.00$	$2.092 \pm.03$	$\mathbf{2.034} \pm.02$
200	1k	$2.152 \pm.00$	$2.154 \pm.00$	$2.097 \pm.00$	$2.168 \pm.00$	$2.168 \pm.00$	$1.995 \pm.02$	$\mathbf{1.945} \pm.01$
250	1.5k	$2.158 \pm.01$	$2.163 \pm.00$	$2.401 \pm.00$	$2.229 \pm.00$	$2.196 \pm.00$	$2.071 \pm.05$	$\mathbf{1.962} \pm.02$
	n|		*Synth. data with increasing sample size and dimensions fixed at $d = 100$*					
	100	$2.194 \pm.03$	$2.203 \pm.04$	$2.262 \pm.00$	$2.156 \pm.01$	$2.127 \pm.00$	$2.476 \pm.18$	$\mathbf{1.955} \pm.03$
	250	$2.238 \pm.00$	$2.244 \pm.01$	$2.150 \pm.00$	$2.233 \pm.01$	$2.232 \pm.00$	$2.109 \pm.02$	$\mathbf{2.032} \pm.01$
	500	$2.119 \pm.01$	$2.029 \pm.01$	$2.395 \pm.00$	$2.230 \pm.00$	$2.118 \pm.00$	$2.092 \pm.04$	$\mathbf{2.008} \pm.00$
	1k	$2.183 \pm.00$	$2.231 \pm.00$	$2.145 \pm.00$	$2.219 \pm.00$	$2.199 \pm.00$	$2.059 \pm.01$	$\mathbf{1.987} \pm.02$
	1.5k	$2.174 \pm.00$	$2.213 \pm.00$	$2.083 \pm.00$	$2.229 \pm.00$	$2.199 \pm.00$	$2.048 \pm.01$	$\mathbf{2.007} \pm.01$
	n|		*Twins ($d = 48$) with increasing sample size*					
	500	$1.092 \pm.11$	1.758 ± 1.1	$1.011 \pm.00$	$1.006 \pm.00$	$1.015 \pm.00$	$0.580 \pm.03$	$\mathbf{0.145} \pm.00$
	1k	$1.015 \pm.00$	$0.963 \pm.00$	$1.010 \pm.00$	$1.004 \pm.00$	$1.010 \pm.00$	$0.549 \pm.04$	$\mathbf{0.161} \pm.00$
	1.5k	$1.014 \pm.00$	$0.965 \pm.00$	$1.005 \pm.00$	$1.006 \pm.00$	$1.012 \pm.00$	$0.546 \pm.04$	$\mathbf{0.154} \pm.00$
	2k	$1.013 \pm.00$	$0.957 \pm.00$	$1.009 \pm.00$	$1.007 \pm.00$	$1.013 \pm.00$	$0.579 \pm.03$	$\mathbf{0.157} \pm.00$
	2.5k	$1.007 \pm.00$	$0.951 \pm.00$	$1.002 \pm.00$	$1.006 \pm.00$	$1.006 \pm.00$	$0.542 \pm.04$	$\mathbf{0.153} \pm.00$
	n|		*IHDP ($d = 25$) with increasing sample size*					
	100	$6.266 \pm.34$	$\mathbf{1.833} \pm.02$	$5.295 \pm.00$	$5.276 \pm.02$	$5.448 \pm.04$	3.580 ± 1.2	$2.444 \pm.73$
	250	$6.243 \pm.06$	$\mathbf{1.624} \pm.04$	$5.180 \pm.00$	$6.145 \pm.07$	$5.564 \pm.02$	$2.811 \pm.73$	$1.729 \pm.19$
	500	$6.995 \pm.05$	$1.951 \pm.01$	$5.293 \pm.00$	$5.988 \pm.06$	$6.469 \pm.15$	$2.671 \pm.14$	$\mathbf{1.635} \pm.09$

A.2 CATE learners with different regression models, and different data

Consider Tables 4-5-6, where we report the PEHE given the same experimental setup as we have in Table 1; for additional data (Infant Health Development Program (IHDP) [MacDorman and Atkinson, 1999]), and three additional regression models (`PowerTransform Regression` [Yeo and Johnson, 2000], `Polynomial Regression`, and `Ridge Regression`, respectively). From our results we learn that our EBM is agnostic to the choice of regression model, and is versatile enough to also perform well given other data. These results are promising and should give some assurance regarding our method before application in practice. As we have in Table 1, we ran each CATE learner on ten distinct representations, given different folds of the data, and averaged the results. Note that we have not specifically optimised the EBM's hyperparameters for these different regression models, but rather kept them as they were in Table 1 (actual hyperparameter values are reported in Table 8). We also include a "complete" version of Table 1 in Table 7, where we include results on IHDP as an additional dataset. Note that these results are in line with those reported earlier using different regression models.

Yao Zhang*, Jeroen Berrevoets*, Mihaela van der Schaar

Table 4: **Results on (semi-)synthetic data (Twins & IHDP) with `PowerTransform` Regression.** We report for the same configuration as in Table 1. Results are averaged over ten runs with ("✓"), and without ("✗") the same representations used in Table 1.

Methods	X-Learner		DR-Learner		T-Learner		R-Learner	
EBM	✗	✓	✗	✓	✗	✓	✗	✓
d n	*Synth. data with increasing sample size and increasing dimensions*							
50 100	2.267 ±.00	**2.010** ±.03	5.593 ±2.2	**2.015** ±.05	2.455 ±.00	**2.011** ±.03	53.17 ±5.2	**11.16** ±3.9
100 250	2.754 ±.00	**2.019** ±.01	3.963 ±.24	**2.027** ±1.2	2.798 ±.00	**2.020** ±.01	60.10 ±4.2	**10.70** ±3.9
150 500	2.575 ±.00	**2.002** ±.01	2.986 ±.08	**2.001** ±.01	2.595 ±.00	**2.001** ±.01	49.78 ±4.2	**12.31** ±1.8
200 1k	2.197 ±.00	**1.952** ±.00	2.293 ±.03	**1.941** ±.01	2.202 ±.00	**1.951** ±.01	42.76 ±3.9	**2.838** ±.64
250 1.5k	2.288 ±.00	**1.966** ±.04	2.410 ±.04	**1.979** ±.03	2.295 ±.00	**1.968** ±.04	42.03 ±3.6	**2.535** ±.15
n	*Synth. data with increasing sample size and dimensions fixed at $d = 100$*							
100	2.150 ±.00	**1.964** ±.02	32.63 ±13	**2.147** ±.15	2.289 ±.00	**1.973** ±.03	58.80 ±5.1	**5.713** ±2.2
250	2.754 ±.00	**2.019** ±.01	3.963 ±.24	**2.027** ±1.2	2.798 ±.00	**2.020** ±.01	60.10 ±4.2	**10.70** ±3.9
500	2.150 ±.00	**2.029** ±.02	2.319 ±.06	**2.006** ±.05	2.160 ±.00	**2.028** ±.03	41.85 ±3.4	**3.884** ±.79
1k	2.053 ±.00	**1.986** ±.02	2.102 ±.01	**1.989** ±.01	2.057 ±.00	**1.987** ±.02	39.49 ±3.2	**2.637** ±.20
1.5k	2.008 ±.00	**1.999** ±.02	2.354 ±.01	**1.999** ±.52	2.008 ±.00	**2.000** ±.02	37.17 ±2.9	**3.949** ±.77
n	*Twins $(d = 48)$ with increasing sample size*							
500	0.203 ±.00	**0.187** ±.06	4.383 ±.22	**0.185** ±.02	**0.204** ±.00	0.248 ±.24	300.0 ±16.	**2.176** ±.89
1k	0.177 ±.00	**0.169** ±.03	0.194 ±.01	**0.163** ±.01	0.177 ±.00	**0.159** ±.02	31.39 ±2.5	**1.029** ±1.1
1.5k	0.169 ±.00	**0.154** ±.00	**0.172** ±.01	0.183 ±.08	0.169 ±.00	**0.155** ±.00	31.23 ±2.3	**0.459** ±.15
2k	0.167 ±.00	**0.161** ±.00	0.168 ±.00	**0.163** ±.00	0.168 ±.00	**0.161** ±.00	29.95 ±2.4	**0.629** ±.30
2.5k	0.169 ±.00	**0.162** ±.00	0.170 ±.00	**0.163** ±.00	0.169 ±.00	**0.162** ±.00	29.79 ±2.4	**0.439** ±.19
n	*IHDP $(d = 25)$ with increasing sample size*							
100	1.814 ±.01	**1.502** ±.03	3.755 ±.72	**1.637** ±.12	1.845 ±.00	**1.507** ±.05	35.77 ±6.3	**21.36** ±17.
250	1.713 ±.00	**1.598** ±.04	1.837 ±.09	**1.653** ±.13	1.727 ±.00	**1.593** ±.03	11.71 ±1.5	**9.552** ±4.0
500	1.603 ±.00	**1.554** ±.02	1.672 ±.05	**1.571** ±.03	1.627 ±.00	**1.556** ±.02	23.45 ±2.5	**15.19** ±3.9

Table 5: **Results on (semi-)synthetic data (Twins & IHDP) using `Polynomial` Regression.** We report for the same configuration as in Table 1. Results are averaged over ten runs with ("✓"), and without ("✗") the same representations used in Table 1.

Methods	X-Learner		DR-Learner		T-Learner		R-Learner	
EBM	✗	✓	✗	✓	✗	✓	✗	✓
d n	*Synth. data with increasing sample size and increasing dimensions*							
50 100	**2.095** ±.00	2.089 ±.09	75.12 ±26.	**2.500** ±.44	2.124 ±.00	**2.095** ±.09	110.8 ±7.1	**26.61** ±7.6
100 250	2.109 ±.00	**2.026** ±.03	11.67 ±1.9	**2.138** ±.30	2.168 ±.00	**2.028** ±.03	50.45 ±4.4	**10.72** ±4.0
150 500	2.048 ±.00	**2.008** ±.02	8.668 ±1.5	**2.006** ±.01	2.142 ±.00	**2.008** ±.02	44.29 ±3.4	**12.35** ±1.8
200 1k	1.964 ±.00	**1.949** ±.02	7.278 ±1.3	**1.949** ±.02	2.088 ±.00	**1.949** ±.02	42.04 ±3.4	**2.973** ±.57
250 1.5k	1.945 ±.00	**1.945** ±.03	6.764 ±1.2	**1.979** ±.03	2.109 ±.00	**1.945** ±.04	41.71 ±3.5	**2.555** ±.18
n	*Synth. data with increasing sample size and dimensions fixed at $d = 100$*							
100	2.009 ±.00	**1.987** ±.09	157.1 ±50.	**2.176** ±.13	2.038 ±.00	**1.992** ±.09	54.81 ±4.3	**6.612** ±3.5
250	2.109 ±.00	**2.019** ±.01	11.67 ±1.9	**2.028** ±.02	2.168 ±.00	**2.019** ±.01	50.46 ±4.4	**9.623** ±3.0
500	**1.897** ±.00	2.055 ±.05	8.020 ±1.7	**2.003** ±.00	**2.007** ±.00	2.051 ±.05	43.56 ±3.9	**3.758** ±.82
1k	2.210 ±.00	**1.995** ±.02	5.871 ±.90	**2.289** ±.88	2.338 ±1.4	**1.996** ±.02	40.38 ±3.1	**2.965** ±.69
1.5k	2.341 ±.00	**2.002** ±.02	4.637 ±.75	**2.156** ±.47	2.474 ±.00	**2.000** ±.02	38.31 ±2.9	**3.945** ±.76
n	*Twins $(d = 48)$ with increasing sample size*							
500	0.345 ±.00	**0.155** ±.00	4.538 ±1.4	**0.158** ±.00	0.377 ±.00	**0.155** ±.00	116.0 ±14.	**0.847** ±.19
1k	0.486 ±.00	**0.149** ±.00	1.747 ±.33	**0.157** ±.01	0.529 ±.00	**0.149** ±.00	78.22 ±7.3	**0.482** ±.11
1.5k	0.455 ±.00	**0.153** ±.00	1.453 ±.40	**0.186** ±.06	0.481 ±.00	**0.153** ±.00	142.3 ±21.	**0.395** ±.13
2k	0.426 ±.00	**0.159** ±.00	1.109 ±.32	**0.162** ±.00	0.451 ±.00	**0.159** ±.00	38.13 ±4.2	**0.655** ±.33
2.5k	0.403 ±.00	**0.159** ±.00	0.921 ±.14	**0.163** ±.01	0.418 ±.00	**0.159** ±.00	33.39 ±2.8	**0.459** ±.19
n	*IHDP $(d = 25)$ with increasing sample size*							
100	1.608 ±.02	**1.565** ±.25	8.076 ±2.4	**1.956** ±.64	1.944 ±.00	**1.542** ±.18	24.53 ±5.1	**11.52** ±8.4
250	2.335 ±.00	**1.637** ±.02	8.607 ±.82	**1.964** ±.49	2.219 ±.00	**1.627** ±.03	35.37 ±3.8	**18.13** ±7.8
500	2.177 ±.00	**1.536** ±.00	4.216 ±.35	**1.739** ±.36	2.233 ±.00	**1.535** ±.00	26.06 ±4.5	**10.18** ±5.6

Table 6: **Results on (semi-)synthetic data (Twins & IHDP) with `Ridge Regression`.** We report for the same configuration as in Table 1. Results are averaged over ten runs with ("✓"), and without ("✗") the same representations used in Table 1.

Methods	X-Learner		DR-Learner		T-Learner		R-Learner	
EBM	✗	✓	✗	✓	✗	✓	✗	✓
d n	*Synth. data with increasing sample size and increasing dimensions*							
50 100	2.373 ±.00	**2.001** ±.02	10.53 ±4.8	**2.028** ±.06	2.471 ±.00	**1.997** ±.02	53.24 ±5.3	**11.33** ±3.9
100 250	2.802 ±.00	**2.021** ±.01	8.769 ±1.5	**2.041** ±.05	2.871 ±.00	**2.021** ±.01	130.6 ±75.	**22.30** ±8.3
150 500	2.581 ±.00	**2.001** ±.01	3.074 ±.10	**2.001** ±.01	2.601 ±.00	**2.001** ±.01	50.13 ±4.1	**12.31** ±1.8
200 1k	2.187 ±.00	**1.942** ±.01	2.304 ±.03	**1.941** ±.01	2.192 ±.00	**1.941** ±.01	42.77 ±3.8	**2.839** ±.62
250 1.5k	2.270 ±.00	**1.958** ±.02	2.412 ±.04	**1.977** ±.03	2.274 ±.00	**1.957** ±.02	42.03 ±3.5	**2.511** ±.17
n	*Synth. data with increasing sample size and dimensions fixed at $d=100$*							
100	2.124 ±.00	**1.946** ±.03	78.12 ±24.	**2.145** ±.11	2.272 ±.00	**1.948** ±.04	58.24 ±4.9	**5.346** ±1.9
250	2.802 ±.00	**2.018** ±.00	4.489 ±.26	**2.025** ±.01	2.871 ±.00	**2.019** ±.01	60.27 ±4.1	**9.607** ±3.0
500	2.152 ±.00	**2.059** ±.04	2.373 ±.08	**2.003** ±.05	2.164 ±.00	**2.056** ±.04	42.13 ±3.5	**3.755** ±.83
1k	2.062 ±.00	**1.984** ±.02	2.109 ±.01	**2.022** ±.09	2.064 ±.00	**1.983** ±.02	39.49 ±3.2	**2.608** ±.19
1.5k	2.013 ±.00	**2.003** ±.02	2.052 ±.01	**2.398** ±1.2	2.014 ±.00	**2.001** ±.02	37.19 ±2.9	**3.954** ±.77
n	*Twins ($d=48$) with increasing sample size*							
500	0.182 ±.00	**0.151** ±.00	1.161 ±.20	**0.168** ±.02	0.183 ±.00	**0.151** ±.00	134.6 ±12.	**0.842** ±.22
1k	0.196 ±.00	**0.159** ±.00	0.261 ±.02	**0.172** ±.01	0.196 ±.00	**0.159** ±.00	58.70 ±4.4	**0.455** ±.14
1.5k	0.166 ±.00	**0.156** ±.00	0.171 ±.01	**0.159** ±.00	0.166 ±.00	**0.156** ±.00	29.72 ±2.4	**0.415** ±.14
2k	0.163 ±.00	**0.153** ±.00	0.319 ±.04	**0.157** ±.00	0.163 ±.00	**0.153** ±.00	176.0 ±13.	**0.601** ±.30
2.5k	0.169 ±.00	**0.162** ±.00	0.321 ±.04	**0.164** ±.00	0.169 ±.00	**0.162** ±.00	207.8 ±17.	**0.479** ±.19
n	*IHDP ($d=25$) with increasing sample size*							
100	1.807 ±.00	**1.673** ±.02	5.933 ±1.2	**2.456** ±.47	1.739 ±.00	**1.675** ±.02	23.56 ±3.2	**14.19** ±9.3
250	1.659 ±.00	**1.579** ±.03	2.883 ±.12	**2.426** ±.09	1.693 ±.00	**1.577** ±.03	11.01 ±2.0	**7.985** ±3.1
500	1.625 ±.00	**1.614** ±.01	1.819 ±.16	**1.673** ±.09	1.641 ±.00	**1.610** ±.02	6.614 ±.59	**5.602** ±1.6

Table 7: **Copy of Table 1 with additional data (IHDP).** To save space, we include the "complete" table of our main experiment here, in our supplemental material. The content of this table is exactly the same as in Table 1, except for added results on IHDP (bottom block).

Methods	X-Learner		DR-Learner		T-Learner		R-Learner	
EBM	✗	✓	✗	✓	✗	✓	✗	✓
d n	*Synth. data with increasing sample size and increasing dimensions*							
50 100	2.309 ±.00	**1.994** ±.02	4.594 ±.56	**2.017** ±.04	2.441 ±.00	**1.993** ±.01	3.194 ±.26	**1.982** ±.04
100 250	2.779 ±.00	**2.018** ±.01	4.056 ±.32	**2.154** ±.39	2.838 ±.00	**2.019** ±.01	3.702 ±.23	**2.018** ±.01
150 500	2.618 ±.00	**2.000** ±.01	3.030 ±.12	**2.001** ±.01	2.641 ±.00	**2.000** ±.01	2.877 ±.08	**2.000** ±.01
200 1k	2.185 ±.00	**1.940** ±.01	2.283 ±.02	**1.941** ±.01	2.189 ±.00	**1.939** ±.01	2.271 ±.01	**1.940** ±.01
250 1.5k	2.267 ±.00	**1.949** ±.02	2.427 ±.01	**1.976** ±.00	2.271 ±.00	**1.948** ±.01	2.436 ±.02	**1.949** ±.02
n	*Synth. data with increasing sample size and dimensions fixed at $d=100$*							
100	2.134 ±.00	**1.927** ±.01	24.61 ±9.9	**2.096** ±.09	2.279 ±.00	**1.929** ±.01	3.192 ±.13	**1.925** ±.01
250	2.779 ±.00	**2.018** ±.01	4.056 ±.32	**2.154** ±.39	2.838 ±.00	**2.019** ±.01	3.702 ±.23	**2.018** ±.01
500	2.155 ±.00	**2.056** ±.02	2.334 ±.07	**2.273** ±.67	2.166 ±.00	**2.053** ±.02	2.271 ±.05	**2.056** ±.02
1k	2.059 ±.00	**1.964** ±.02	2.105 ±.01	**2.016** ±.16	2.061 ±.00	**1.964** ±.02	2.086 ±.01	**1.965** ±.02
1.5k	2.013 ±.00	**1.998** ±.02	2.043 ±.01	**1.998** ±.02	2.014 ±.00	**1.998** ±.02	2.024 ±.01	**1.991** ±.02
n	*Twins ($d=48$) with increasing sample size*							
500	0.214 ±.00	**0.144** ±.00	0.236 ±.04	**0.182** ±.05	0.221 ±.00	**0.145** ±.00	0.222 ±.02	**0.145** ±.00
1k	0.294 ±.00	**0.162** ±.00	0.348 ±.12	**0.173** ±.03	0.301 ±.00	**0.162** ±.01	0.532 ±.11	**0.161** ±.00
1.5k	0.165 ±.00	**0.154** ±.00	0.189 ±.06	**0.159** ±.01	0.165 ±.00	**0.154** ±.00	0.172 ±.01	**0.154** ±.00
2k	0.167 ±.00	**0.156** ±.00	0.197 ±.03	**0.159** ±.00	0.167 ±.00	**0.156** ±.00	0.222 ±.05	**0.157** ±.00
2.5k	0.297 ±.00	**0.153** ±.00	0.390 ±.19	**0.156** ±.00	0.297 ±.00	**0.153** ±.00	0.358 ±.22	**0.153** ±.00
n	*IHDP ($d=25$) with increasing sample size*							
100	3.369 ±.36	**1.783** ±.22	6.484 ±5.7	**2.329** ±.67	6.138 ±.50	**2.028** ±.65	22.57 ±4.3	**16.43** ±9.1
250	47.29 ±.64	**2.788** ±1.2	1.899 ±.10	**1.689** ±.26	51.76 ±.64	**2.963** ±1.3	72.20 ±9.4	**54.16** ±22.
500	2.176 ±.25	**1.532** ±.01	1.681 ±.04	**1.554** ±.02	4.361 ±.35	**1.552** ±.03	8.531 ±.94	**4.661** ±.75

Table 8: **Chosen hyperparameters for Table 1.** We performed hyperparamter sweeps for each setup using a Bayesian optimisation scheme [Biewald, 2020]. Our searched ranges are reported in Table 9. We have rounded continuous hyperparameters to user-friendly values (as they are sampled from continuous distributions during optimization). Twins settings were also used in Figure 2, but with a fixed $k = 5$ for both AE and EBM. Each integer (separated by a dash) in "Architecture" indicates layer width; "20-20" thus means a neural network with two hidden layers, each of width 20.

Setup		b	k	Architecture	Perturbation prob.
d	n|			Synth. data, increasing dim	
50	100|	10	3	20-20-20	0.20
100	250|	10	4	20-20-20	0.50
150	500|	5	3	20-20	0.20
200	1k|	3	15	20-20-20-20	0.50
250	1.5k|	3	20	20-20-20	0.50
	n|			Synth. data, fixed dim (d=100)	
	100|	5	15	20-20-20-20-20-20	0.20
	250|	10	4	20-20-20	0.50
	500|	3	10	20-20-20-20	0.50
	1k|	3	20	20-20	0.35
	1.5k|	3	10	20-20	0.30
	n|			Twins, increasing n	
	500|	5	15	20-20-20-20-20-20	0.45
	1k|	5	16	20-20-20-20-20-20	0.55
	1.5k|	5	16	20-20-20-20-20-20	0.55
	2k|	4	14	20-20-20-20-20-20	0.55
	2.5k|	4	12	20-20-20-20-20-20	0.50
	n|			IHDP, increasing n	
	100|	1	5	36-36-36-36-36-36	0.45
	250|	1	5	36-36-36-36-36-36	0.45
	500|	1	5	36-36-36-36-36-36	0.45

Table 9: **Ranges for hyperparameter sweeps, for Table 8.** For each setup: (I) Synth. data, increasing dim, (II) Synth. data, fixed dim (d=100) (III) Twins, increasing dim, and (IV) IHDP, increasing dim; we used a Bayesian optimization (BO) scheme to find our selected hyperparameters. We chose BO as training representations can get expensive. In or BO setup, we maximized the loss (7) on a (20%) validation-set.

Setup|	b	k	# layers	Perturbation prob.
(I)	$\mathcal{U}(1; 2; ...; 10)$	$\mathcal{U}(3; 4; ...; 25)$	$\mathcal{U}(2; 3; 4; 5; 6)$	$\mathcal{U}(0.2; 0.8)$
(II)	$\mathcal{U}(1; 2; ...; 10)$	$\mathcal{U}(3; 4; ...; 25)$	$\mathcal{U}(2; 3; 4; 5; 6)$	$\mathcal{U}(0.2; 0.8)$
(III)	$\mathcal{U}(1; 2; ...; 10)$	$\mathcal{U}(3; 4; ...; 25)$	$\mathcal{U}(2; 3; 4; 5; 6)$	$\mathcal{U}(0.2; 0.8)$
(IV)	$\mathcal{U}(1; 2; ...; 10)$	$\mathcal{U}(3; 4; ...; 25)$	$\mathcal{U}(2; 3; 4; 5; 6)$	$\mathcal{U}(0.2; 0.8)$

A.3 Hyperparameters

We report our chosen hyperparameters for each sample-size in Table 8. We found these values through a Bayesian optimization scheme [Biewald, 2020]. The used ranges are reported in Table 9. As an insight, we noticed that the architecture and amount of noisy samples made little difference to performance in PEHE. The perturbation probability, and the value of k *did* make a difference, especially in larger sample sizes. Each experiment was performed on an Nvidia `GeForce RTX 2080 Ti` GPU, and 6 Intel `i5-8600K (3.60GHz)` CPUs. In some instances we ran independent experiments on a duplicate system.

A.4 Details on benchmarked CATE estimators

We use EconML [Microsoft Research, 2019][5] to evaluate the CATE learners. We keep the hyperparameters for each CATE learner as their default, except for the regression models (which by default are linear). In Table 1 we replace each regressor by a `KernelRidge` regressor, and each classifier by a support vector machine (`SVC`); both implemented by Pedregosa et al. [2011]. Results on CATE learners with alternative models are in Appendix A.2.

---

[5]EconML is available open-source under an MIT License. Please find all details on their repository, `https://github.com/microsoft/EconML`.

B   Proofs

B.1   Proof of Proposition 3.1

Proof. The sufficiency of f_θ is obtained immediately by applying Fisher–Neyman factorization theorem to (3). Consider another sufficient statistics $\tilde{u} : \mathcal{X} \to \mathcal{U}$. By the factorization theorem, we can factorize the distribution as $p_{\theta,j}(x) = \tilde{h}(x)\tilde{g}(\tilde{u}(x), \beta_j)$ for some functions $\tilde{h}$ and $\tilde{g}$. Given any x and x' such that $\tilde{u}(x) = \tilde{u}(x')$, i.e., $\tilde{g}(\tilde{u}(x), \beta_j) = \tilde{g}(\tilde{u}(x'), \beta_j)$, we have $\exp\left[-\beta_j^\top (f_\theta(x) - f_\theta(x'))\right] = \frac{p_{\theta,j}(x)}{p_{\theta,j}(x')} = \frac{\tilde{h}(x)}{\tilde{h}(x')}$.

The first equality uses two facts of the standard EBM: (1) $h(x)$ is constant ($h(x) = 1$), and (2) $Z_{\theta,j}$ is for fixed β_j. The ratio on the right-hand side does not depend on β_j, which holds if and only if $f_\theta(x) = f_\theta(x')$. That is, for any sufficient statistics $\tilde{u}$ and any x, x', $\tilde{u}(x) = \tilde{u}(x')$ implies that $f_\theta(x) = f_\theta(x')$ and hence that $f_\theta(x)$ is a function of $\tilde{u}(x)$. Then f_θ is a minimal sufficient statistic. □

B.2   Proof of Proposition 3.3

Proof. Consider two different parameter values θ and $\tilde{\theta}$ such that $p_{\theta,j}(x) = p_{\tilde{\theta},j}(x)$. Using the expression (3) and applying logarithm to both sides,

$$\beta_j^\top f_\theta(x) = \beta_j^\top f_{\tilde{\theta}}(x) + \log \frac{Z_{\tilde{\theta},j}}{Z_{\theta,j}}. \tag{8}$$

By concatenating the last equation for all $j \in [k]$, we have

$$B^\top f_\theta(x) = B^\top f_{\tilde{\theta}}(x) + G \tag{9}$$

where $G = \left(\log \frac{Z_{\tilde{\theta},j}}{Z_{\theta,j}} : j \in [k]\right)$ is a k-dimensional vector. By definition, $BB^\top = I_{k \times k}$. Then multiplying the two side of (9) by B proves eq. (5),

$$f_\theta(x) = f_{\tilde{\theta}}(x) + C \text{ for any } x \in \mathcal{X} \text{ and } C = BG. \tag{10}$$

Reversely, multiplying two sides of (10) by $\beta_j^\top$, we obtain eq. (8),

$$\beta_j^\top f_\theta(x) = \beta_j^\top f_{\tilde{\theta}}(x) + \beta_j^\top BG = \beta_j^\top f_{\tilde{\theta}}(x) + G_j = \beta_j^\top f_{\tilde{\theta}}(x) + \log \frac{Z_{\tilde{\theta},j}}{Z_{\theta,j}}$$

Then multiplying -1 and applying $\exp(\cdot)$ to both sides,

$$p_{\theta,j}(x) = Z_{\theta,j}^{-1} \exp\left[-\beta_j^\top f_\theta(x)\right] = Z_{\tilde{\theta},j}^{-1} \exp\left[-\beta_j^\top f_{\tilde{\theta}}(x)\right] = p_{\tilde{\theta},j}(x).$$

□

The orthogonality of B is also the key to prove the universal approximation capability of our partially randomized EBM in Proposition 3.4; see the next section for more details.

B.3   Proof of Proposition 3.4

Theorem B.1 (Stone-Weierstrass; Theorem 4.45 in [Folland, 1999])**.** *Suppose $\mathcal{X} = [0,1]^d$, and that $\mathcal{P}$ is a class of functions satisfying the following conditions:*

1. *Every $p_X \in \mathcal{P}$ is continuous.*

2. *For every $x \in \mathcal{X}$, there exists $p_X \in \mathcal{P}$ such that $p_X(x) \neq 0$.*

3. *For every $x, x' \in \mathcal{X}$ such that $x \neq x'$, there exists $p_X \in \mathcal{P}$ such that $p_X(x) \neq p_X(x')$.*

4. *$\mathcal{F}$ is closed under multiplication and under vector space operations.*

Then for every continuous density function $g : \mathcal{X} \to \mathbb{R}^+$ and $\epsilon > 0$, there exists $p_X \in \mathcal{P}$ s.t. $\sup_{x \in \mathcal{X}} |g(x) - p_X(x)| \leq \epsilon$.

The original Stone-Weierstrass theorem works for any $g : \mathcal{X} \to \mathbb{R}$. Here we modify the codomain of g to be $\mathbb{R}^+$ because we only consider estimating density functions. The Stone-Weierstrass theorem works for any $\mathcal{X} \subseteq \mathbb{R}^d$. Choosing $\mathcal{X} = [0,1]^d$ is common practice in the literature of approximation theory and nonparametric regression. It is used to simplify the notation in the proof. The proof idea holds for general $\mathcal{X} \subseteq \mathbb{R}^d$.

Proof. Suppose f_θ is a linear neural network in the model (3), we have

$$p_{\theta,j} \in \mathcal{P}(\beta_j) := \left\{ x \to Z_{\theta,j}^{-1} \exp\left[-\beta_j^\top \theta x \right] : \theta \in \mathbb{R}^{k \times d} \right\}$$

The first condition in Theorem B.1 is satisfied directly by definition. If θ is a zero matrix, then $p_{\theta,j}(x) \neq 0$, which means the second condition in Theorem B.1 holds. Given any $x_1, x_2 \in \mathcal{X}$ such that $x_1 \neq x_2$, we can always find a exponential function along the line between x_1 and x_2. Because the exponential function $p_{\theta,j} \in \mathcal{P}(\beta_j)$ is nonlinear, we have $p_{\theta,j}(x_1) \neq p_{\theta,j}(x_2)$. Formally, we let $\theta = -\left(\sum_{j'=1}^k \beta_{j'} \right)(x_1 - x_2)^\top$, then

$$-\beta_j^\top \theta = 1 \cdot (x_1 - x_2)^\top = (x_1 - x_2)^\top,$$

The first equality uses the fact that β_j is a column of the random orthogonal matrix B. Then,

$$p_{\theta,j}(x) = Z_{\theta,j}^{-1} \exp\left[(x_1 - x_2)^\top x \right].$$

Because $x_1 \neq x_2$, we have

$$\frac{p_{\theta,j}(x_1)}{p_{\theta,j}(x_2)} = \frac{\exp\left[x_1^\top x_1 - x_2^\top x_1 \right]}{\exp\left[x_1^\top x_2 - x_2^\top x_2 \right]} = \exp\left[-(x_1 - x_2)^\top (x_1 - x_2) \right] > 0,$$

which proves that that the third condition in Theorem B.1 holds. The space $\mathcal{P}(\beta_j)$ is closed under vector space operations by definition. It is also closed under multiplication. By the property of exponential functions, $\exp\left[-\beta_j^\top \theta x_1 \right] \cdot \exp\left[-\beta_j^\top \theta x_2 \right] = \exp\left[-\beta_j^\top \theta (x_1 + x_2) \right]$, it is straightforward to show that given some functions p's $\in \mathcal{P}(\beta_j)$, the multiplication of two different linear combinations of p's is still in the span of p's. Thus, $\mathcal{P}(\beta_j)$ satisfies the fourth condition in Theorem B.1. □

Although the derivation is based on a linear network f_θ for simplicity, it does not mean we should use a linear network in practice. In recent years, it has been shown theoretically and empirically that overparametrization in deep neural networks allows gradient methods to find interpolating solutions; these methods implicitly impose regularization; overparametrization leads to benign overfitting, that is, accurate predictions (i.e. better generalization) on the testing data despite overfitting training data; see [Bartlett et al., 2021] for a detailed review of the recent theoretical analysis on overparameterized models.

B.4 Proof of Proposition 4.1

Proof. As discussed in the main paper, both MLE and NCE are M-estimators. The proof consists of two steps: (1) we show $q_{\theta,j}(a \mid V_i)$ with $p_{\theta,j}(x) = p_X(x)$ is a maximizer of $\mathcal{L}_{\infty,j}(\theta)$, then (2) we show the standard conditions of consistent M-estimators hold for $\mathcal{L}_{n,j}(\theta)$. We assume the sample splitting always keep the individuals in the same folds as $n \to \infty$. This can be achieved by keeping the existing individuals in the same folds and randomly assign a new individual to a fold as $n \to \infty$. For large n, each fold will have roughly the same number of individuals, and $n_j = |\mathcal{I}_j| \to \infty$ for every $j \in [k]$ as $n \to \infty$. The assumption does not affect our method in practice because we only consider the number of individuals n observed in our dataset.

Step 1. Recall that for every $i \in [n]$, $\bar{X}_i = (X_i, \tilde{X}_{i1}, \ldots, \tilde{X}_{ib}) \sim p_X(x) \prod_{a=1}^b \tilde{p}_{\tilde{X}|X}(\tilde{x} \mid x)$. We randomly permute the columns of $\bar{X}_i$ and let $V_i = (V_{i,1}, \cdots V_{i,b+1})$ be the permuted $\bar{X}_i$. Each column of V_i has equal probability $(b+1)^{-1}$ for being the clean sample X_i. The variable $W_i \in \{0,1\}^{b+1}$ indicate which column of V_i is X_i. Here, we define a categorical variable $S_i \in [b+1]$ such that $S_i = a$ if $W_{ia} = 1$. We know that

$$p_{S_i}(a) = 1/(b+1), \ \forall a \in [b+1].$$

We define the marginal distribution of $v = (v_c : c \in [b+1])$ as

$$\Lambda_j(v) = \sum_{a=1}^{b+1} p_S(a) p_{V|S}(v|a) = \sum_{a=1}^{b+1} (b+1)^{-1} p_X(v_a) \tilde{p}_{-a}(v)$$

where $\tilde{p}_{-a}(v) = \prod_{a \in [b+1]:a' \neq a} p_{\tilde{X}|X}(v_{a'} \mid v_a)$. We define the posterior probability of $S = a$ as

$$p_{S|V}(a \mid v) = \frac{(b+1)^{-1}p_X(v_a)\tilde{p}_{-a}(v)}{\sum_{c=1}^{b+1}(b+1)^{-1}p_X(v_c)\tilde{p}_{-c}(v)}$$

This corresponds to the posterior probability $q_{\theta,j}(a \mid v)$ based on the model $p_{\theta,j}(x)$ in (6).

As $n \to \infty$, i.e., $n_j \to \infty$, the objective function in (7) is given by

$$\mathcal{L}_{\infty,j}(\theta) = \int \Lambda_j(v) \left[\sum_{a=1}^{b+1} p_{S|V}(a \mid v) \log q_{\theta,j}(a \mid v) \right] dv$$

Because $\Lambda_j(v) > 0$ and Lemma B.2, $\mathcal{L}_{\infty,j}(\theta)$ is maximized when

$$q_{\theta,j}(a \mid v) = p_{S|V}(a \mid v), \ \forall a \in [b+1]. \tag{11}$$

Suppose $v = (v_{a'} : a' \in [b+1])$ satisfies that $v_{a'} = \xi$ for all $a' \in [b+1] \setminus \{a\}$. Then

$$\tilde{p}_{-c}(v) = \tilde{p}_{\tilde{X}|X}(v_a \mid \xi) \left[p_{\tilde{X}|X}(\xi \mid \xi) \right]^{b-1}, \forall c \in [b+1] \setminus \{a\}$$

and

$$\tilde{p}_{-c}(v) = \tilde{p}_{-c'}(v), \forall c, c' \in [b+1] \setminus \{a\}. \tag{12}$$

We continue to rewrite (11) as

$$\frac{p_{\theta,j}(v_a)\tilde{p}_{-a}(v)}{\sum_{c=1}^{b+1} p_{\theta,j}(v_c)\tilde{p}_{-c}(v)} = \frac{p_X(v_a)\tilde{p}_{-a}(v)}{\sum_{c'=1}^{b+1} p_X(v_{c'})\tilde{p}_{-c'}(v)}$$

$$\frac{p_{\theta,j}(v_a)}{\sum_{c=1}^{b+1} p_{\theta,j}(v_c)\tilde{p}_{-c}(v)} = \frac{p_X(v_a)}{\sum_{c'=1}^{b+1} p_X(v_{c'})\tilde{p}_{-c'}(v)}$$

$$\frac{\sum_{c=1}^{b+1} p_{\theta,j}(v_c)\tilde{p}_{-c}(v)}{p_{\theta,j}(v_a)} = \frac{\sum_{c'=1}^{b+1} p_X(v_{c'})\tilde{p}_{-c'}(v)}{p_X(v_a)}$$

$$\tilde{p}_{-a}(v_a) + \sum_{c \neq a} \frac{p_{\theta,j}(v_c)\tilde{p}_{-a}(v_c)}{p_{\theta,j}(v_a)} = \tilde{p}_{-a}(v_a) + \sum_{c' \neq a} \frac{p_X(v_{c'})\tilde{p}_{-a}(v_{c'})}{p_X(v_a)}$$

$$\frac{p_{\theta,j}(\xi)}{p_{\theta,j}(v_a)} = \frac{p_X(\xi)}{p_X(v_a)}$$

$$\beta_j^\top \left[f_\theta(\xi) - f_\theta(v_a) \right] = \beta_j^\top \left[f_{\theta_0}(\xi) - f_{\theta_0}(v_a) \right],$$

The fifth line is attained by (12). The last line is achieved by the setup of our proposition: $p_{\theta_0,j}(x) = p_X(x)$ for any $\theta_0 \in \Theta_0$. Combing the last equation for all $j \in [k]$ and using the fact that B is a orthogonal matrix, we have

$$B^\top \left[f_\theta(\xi) - f_\theta(v_a) \right] = B^\top \left[f_{\theta_0}(\xi) - f_{\theta_0}(v_a) \right]$$
$$f_\theta(\xi) - f_\theta(v_a) = f_{\theta_0}(\xi) - f_{\theta_0}(v_a)$$
$$f_\theta(v_a) = f_{\theta_0}(v_a) + C(\theta, \theta_0)$$

Then,

$$p_{\theta,j}(x) = Z_{\theta,j}^{-1} \exp \left[-\beta_j^\top f_\theta(x) \right] = \frac{\exp \left[-\beta_j^\top f_{\theta_0}(x) - \beta_j^\top C(\theta, \theta_0) \right]}{\int_{\mathcal{X}} \exp \left[-\beta_j^\top f_{\theta_0}(x) - \beta_j^\top C(\theta, \theta_0) \right] dx} = p_X(x).$$

For any $\hat{\theta} \in \arg\max_{\theta \in \Theta} \mathcal{L}_{\infty,j}(\theta)$, we have $p_{\hat{\theta},j}(x) = p_X(x)$ for any $x \in \mathcal{X}$.

Step 2. Because $\mathcal{X}$ and Θ are compact, the value of each covariate and network parameter is bounded. Then the function we optimize in eq. (7) is

$$g_j(\bar{x}; \theta) = \log q_{\theta,j}(1 \mid \bar{x})$$

is bounded for any $\bar{x} = (x, \tilde{x}_{i1}, \ldots, \tilde{x}_{ib})$. Then we denote $\mathcal{L}_{n,j}(\theta)$ in eq. (7) by $\mathbb{E}_n\left[g_j(\bar{X};\theta)\right] = n_j^{-1}\sum_{i\in\mathcal{I}_j}g_j(\bar{X}_i;\theta)$. Using the uniform law of large number (ULLN) (see [Jennrich, 1969, Theorem 2] and [Newey and McFadden, 1994, Lemma 2.4]), we have

$$\sup_{\theta\in\Theta}\left|\mathbb{E}_n\left[g_j(\bar{X};\theta)\right] - \mathbb{E}\left[g_j(\bar{X};\theta)\right]\right| \xrightarrow{p} 0.$$

Now we change the exact identifiability assumption in [Wooldridge, 2010, Theorem 12.2] and [Newey and McFadden, 1994, Theorem 2.5] to our partial identifiability assumption. Under Proposition 3.4, we suppose that there is a countable subset $\Theta_0 \subset \Theta$ such that for every $\theta_0 \in \Theta_0$, $p_{\theta_0,j}(x)$ gives the same distribution as $p_X(x)$, and any θ_0 is a non-unique maximizer of $\mathbb{E}\left[g_j(\bar{X};\theta)\right]$. We define an open ball with radius equal to $\eta > 0$ for every $\theta_0 \in \Theta_0$. The region inside and outside these open balls is given by

$$\Theta_\eta = \left\{\theta\in\Theta \;\middle|\; \arg\min_{\theta_0\in\Theta_0}\|\theta-\theta_0\|_2 < \eta\right\} \quad \text{and} \quad \Theta_\eta^c = \left\{\theta\in\Theta \;\middle|\; \arg\min_{\theta_0\in\Theta_0}\|\theta-\theta_0\|_2 \geq \eta\right\}.$$

Using the proof of [Newey and McFadden, 1994, Theorem 2.1], $\forall\epsilon > 0$, $\theta_0 \in \Theta_0$ and $\hat{\theta}_n \in \arg\max \mathbb{E}_n\left[g_j(\bar{X};\theta)\right]$, we have with probability approaching to 1:

$$\mathbb{E}\left[g_j(\bar{X};\hat{\theta}_n)\right] > \mathbb{E}\left[g_j(\bar{X};\theta_0)\right] - \epsilon. \tag{13}$$

By the compactness of Θ_η^c and the assumption that f_θ is continuous w.r.t to θ, we have

$$\sup_{\theta\in\Theta_\eta^c}\mathbb{E}\left[g_j(\bar{X};\theta)\right] = \mathbb{E}\left[g_j(\bar{X};\theta^*)\right] < \mathbb{E}\left[g_j(\bar{X};\theta_0)\right] \text{ for some } \theta^* \in \Theta_\eta^c.$$

Thus, by $\epsilon = \mathbb{E}\left[g_j(\bar{X};\theta_0)\right] - \sup_{\theta\in\Theta_\eta^c}\mathbb{E}\left[g_j(\bar{X};\theta)\right]$, it follows from (13) that with probability approaching to 1,

$$\mathbb{E}\left[g_j(\bar{X};\hat{\theta}_n)\right] > \sup_{\theta\in\Theta_\eta^c}\mathbb{E}\left[g_j(\bar{X};\theta)\right] \Rightarrow \hat{\theta}_n \in \Theta_\eta. \tag{14}$$

Since (14) is true for any $\eta > 0$, we have $\hat{\theta}_n \in \Theta_0$ with probability 1 as $n \to \infty$. We note that the same proof holds if we consider the summation of $\mathbb{E}_n\left[g_j(\bar{X};\theta)\right]$ over all $j \in [k]$. Therefore, for any number of noise samples b and $\hat{\theta}_n \in \arg\max_{\theta\in\Theta}\mathcal{L}_n(\theta)$, $\lim_{n\to\infty}\hat{\theta}_n \in \Theta_0$ with probability 1. □

B.5   Supporting Lemma

Lemma B.2. *Suppose* $w = (w_1, \ldots, w_b) > 0$ *and* $\sum_{a=1}^b w_a = 1$,

$$f(\tilde{w}; w) = \sum_{a=1}^b w_a \log\tilde{w}_a \quad \text{subject to} \quad \tilde{w} = (\tilde{w}_1, \ldots, \tilde{w}_b) > 0 \text{ and } \sum_{a=1}^b \tilde{w}_a = 1.$$

Then $f(\tilde{w}; w)$ *is maximized at* $\tilde{w} = w$.

Proof. Suppose $g(\tilde{w}; w) = \sum_{a=1}^b w_a \log\tilde{w}_a + \lambda\left(\sum_{a=1}^b \tilde{w}_a - 1\right)$. We have

$$\frac{\partial g(\tilde{w}; w)}{\partial\tilde{w}_c} = 0 \Rightarrow \tilde{w}_c = -\frac{w_c}{\lambda} \quad \text{and} \quad \frac{\partial g(\tilde{w}; w)}{\partial\lambda} = 0 \Rightarrow \sum_{a=1}^b \tilde{w}_a = 1.$$

Combining both conditions, we have $\sum_{a=1}^b \tilde{w}_a = -\frac{1}{\lambda}\sum_{a=1}^b w_c = -\frac{1}{\lambda} = 1 \Rightarrow \lambda = -1$. Then, $\tilde{w}_c = -\frac{w_c}{-1} \Rightarrow \tilde{w}_c = w_c$. Then by a second-derivative test on the bordered Hessian of $g(\tilde{w}; w)$, we have $\tilde{w} = (\tilde{w}_1, \ldots, \tilde{w}_b) = (w_1, \ldots, w_b)$ is a maximizer of the function $f(\tilde{w}; w)$. □

B.6 The noise distribution $\tilde{p}(\tilde{x} \mid X_i)$

We draw a noise sample from $p_{\tilde{X}|X}(\tilde{x} \mid X_i)$. For each feature X_{is} of the d-dimensional X_i, $s \in [d]$, we sample an independent binary variable $R_{is} \sim \text{Ber}(q)$ to decide if the s-th feature X_{is} will be corrupted. If $R_{is} = 1$, we will corrupt the s-th feature, otherwise not. Overall, the first part of $p_{\tilde{X}|X}(\tilde{x} \mid X_i)$ is given by

$$\prod_{s=1}^{d} q^{R_{is}} (1-q)^{1-R_{is}},$$

where q is the only hyperparameter in $p_{\tilde{X}|X}(\tilde{x} \mid X_i)$. We use the same q for all $j \in [k]$. In Appendix A.3, we describe how q (called perturbation prob.) is selected by validation in our experiments.

Suppose that $R_{is} = 1$. If the s-th feature is continuous, we will corrupt it by adding a white noise E_{is} drawn from a standard normal distribution, i.e.,

$$E_{is} \sim p_{E_s|R_s}(e_s \mid 1) = \mathcal{N}(0,1).$$

If the s-th feature is categorical and takes its value in $\mathcal{X}_s$, we will corrupt it by replacing X_{is} with a uniform sample E_{is} drawn from the same range $\mathcal{X}_s$, i.e.,

$$E_{is} \sim p_{E_s|R_s}(e_s \mid 1) = 1/|\mathcal{X}_s|.$$

Suppose that $R_{is} = 0$. We do not corrupt the s-th feature. That is,

- $E_{is} = 0$ and $p_{E_s|R_s}(0 \mid 0) = 1$ if the s-th feature is continuous and $R_{is} = 0$;

- $E_{is} = X_{is}$ and $p_{E_s|R_s,X_s}(X_{is} \mid 0, X_{is}) = 1$ if the s-th feature is categorical and $R_{is} = 0$.

Overall, the probability $p_{\tilde{X}|X}(\tilde{X}_i \mid X_i)$ is computed using R_{is}, E_{is} and X_{is} for all $s \in [d]$,

$$p_{\tilde{X}|X}(\tilde{X}_i \mid X_i) = \prod_{s=1}^{d} q^{R_{is}} (1-q)^{1-R_{is}} p_{E_s|R_s,X_s}(E_{is} \mid R_{is}, X_{is}).$$

where $p_{E_s|R_s,X_s}(E_{is} \mid R_{is}, X_{is})$ only depends on X_{is} if the s-th feature is categorical and $R_{is} = 0$, otherwise $p_{E_s|R_s,X_s}(E_{is} \mid R_{is}, X_{is}) = p_{E_s|R_s}(E_{is} \mid R_{is})$. The corrupted sample $\tilde{X}_i$ is obtained by either adding E_{is} to X_{is} or replacing X_{is} by E_{is} for every $s \in [d]$. We do not need to consider this step when we compute $\tilde{p}_{\tilde{X}|X}(\tilde{X}_i \mid X_i)$.