End-to-End Balancing for Causal Continuous Treatment-Effect Estimation

Mohammad Taha Bahadori [1] **Eric J. Tchetgen Tchetgen** [1 2] **David E. Heckerman** [1]

Abstract

We study the problem of observational causal inference with continuous treatment. We focus on the challenge of estimating the causal response curve for infrequently-observed treatment values. We design a new algorithm based on the framework of entropy balancing which learns weights that directly maximize causal inference accuracy using end-to-end optimization. Our weights can be customized for different datasets and causal inference algorithms. We propose a new theory for consistency of entropy balancing for continuous treatments. Using synthetic and real-world data, we show that our proposed algorithm outperforms the entropy balancing in accuracy of treatment effect estimation.

1. Introduction

In many applications in business, social, and health sciences, we wish to infer the effect of a continuous treatment such as drug dosage or administration duration on a health outcome variable. Often, several confounding factors are common factors of influencing both treatment and response variable, therefore for accurate causal estimation of the treatment in view, we must appropriately account for their potential impact. Unlike binary treatments, causal inference with continuous treatments is largely understudied and far more challenging than binary treatments. (Galagate, 2016; Ai et al., 2021). This is primarily because continuous treatments induce uncountably many potential outcomes per unit, only one of which is observed for each unit and across units, a sparse coarsening of the underlying information needed to infer causal effects without uncertainty.

Propensity score weighting (Robins et al., 2000; Imai & Van Dyk, 2004), stand-alone or combined with regression-based models to achieve double robustness (Díaz & van der Laan, 2013; Kennedy et al., 2017), has quickly become the state of the art for causal inference. If the weights, inversely proportional to the conditional distribution of the treatment given the confounders, are correctly modeled, the weighted population will appear to come from a randomized study. However, this approach faces several challenges: (1) The weights only balance the confounders in expectation, not necessarily in the given data (Zubizarreta et al., 2011). (2) The weights can be very large for some of units, leading to unstable estimation and uncertain inference. As a possible remedy, entropy balancing (Hainmueller, 2012) estimates the weights such that they balance confounders subject to a measure of dispersion on the weights to prevent extreme weights.

In this work, we note that low-entropy weights do not directly optimize the quality of subsequent weighted regression, and we introduce an alternative approach that does. We propose *End-to-End Balancing* (E2B) to improve the accuracy of the weighted regression used for causal inference. E2B uses end-to-end training to estimate the base weights in the entropy balancing framework. The E2B weights are thus customized for different datasets and causal inference algorithms that are based on weighting. Because we do not know the true treatment response function in real data, we propose a new approach to generate synthetic training datasets for end-to-end training.

To theoretically analyze end-to-end balancing, we define *Generalized Stable Weights* (GSW) for causal inference as a generalization of the stable weights proposed by Robins et al. (2000). We prove that weights learned by entropy balancing for continuous treatments, including E2B weights, are unbiased estimators of generalized stable weights. We also show that E2B weights are asymptotically consistent and efficient estimators of the population weights.

We perform three sets of experiments to demonstrate accuracy improvements by E2B. Two experiments with synthetic data, one with linear and another with non-linear response functions show that the E2B is more accurate than the baseline entropy balancing and inverse propensity score techniques. We also study the impact of mis-specification in synthetic data generation process. In the experiments on real-world data, we qualitatively evaluate the average treatment effect function learned by E2B. We also show that

[1] Amazon.com, Inc. [2] Wharton School of the University of Pennsylvania. Correspondence to: Mohammad Taha Bahadori <bahadorm@amazon.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022. Copyright 2022 by the author(s).

the base weights learned by E2B follow our intuition about up-weighting low frequency treatments.

2. Problem Definition and Related Work

Problem Statement. Suppose we have the triplet of $(\mathbf{x}, \mathrm{a}, \mathrm{y})$, where $\mathbf{x} \in \mathbb{X} \subset \mathbb{R}^r$, $\mathrm{a} \in \mathbb{A} \subset \mathbb{R}$ and $\mathrm{y} \in \mathbb{R}$ denote the confounders, treatments, and response variables, respectively, from an observational causal study. In our continuous treatment setting (Galagate, 2016, Ch. 1.2.6), we denote potential outcomes as $\mathrm{y}^{(a)}$, which means the value of y after intervention in the treatment a and setting its value to a. Given an i.i.d. sample of size n, $\{(\boldsymbol{x}_i, a_i, y_i)\}_{i=1}^n$, our objective is to eliminate the impact of the confounders and identify the average treatment effect function $\mu(a) = \mathbb{E}[\mathrm{y}^{(a)}]$, which is also called the response function. We make the two classic assumptions: (1) Strong ignorability: $\mathrm{y}^{(\mathrm{a})} \perp\!\!\!\perp \mathrm{a} \mid \mathbf{x}$. (i.e., no hidden confounders) and (2) Positivity: $0 < P(\mathrm{a}|\mathbf{x}) < 1$.

General Causal Inference Literature. The literature on causal inference is vast and we refer the reader to the books for the general inquiry (Pearl, 2009; Imbens & Rubin, 2015; Spirtes et al., 2000; Peters et al., 2017). Instead, we focus on reviewing the inference techniques for *continuous* treatments. In particular, we narrow down our focus on propensity score weighting approaches (Robins et al., 2000; Imai & Van Dyk, 2004), because they can either be used alone or combined with the regression algorithms to create doubly robust algorithms.

Causal Inference via Weighting. A popular approach for causal inference is to create a pseudo-population by weighting data points such that in the pseudo-population the confounders and treatments are independent. Thus, regular regression algorithms can estimate the causal response curve using the pseudo-population, which resembles data from randomized trials. Throughout this paper, we will denote the parameters of the pseudo-population with a tilde mark. Multiple forms of propensity scores have been proposed for continuous treatments (Hirano & Imbens, 2004; Imai & Van Dyk, 2004). The commonly-used *stablized weights* (Robins et al., 2000; Zhu et al., 2015) are defined as the ratio of marginal density over the conditional density of the treatments: $sw = {f(a)}/{f(a|\boldsymbol{x})}$.

Problems with Propensity Scores. Zubizarreta et al. (2011) list two challenges with the propensity scores: (1) The weights only balance the confounders in expectation, not necessarily in the given data. (2) The weights can be very large for some of the data points, leading to unstable estimations. The challenges are amplified in the continuous setting because computing the stabilized weights requires correctly choosing two models, one for the marginal and one for the conditional distributions of the treatments. Kang et al. (2007) and Smith & Todd (2005) provide multiple evidence that the propensity score methods can lead to large biases in the estimations. While Robins et al. (2007) propose techniques to fix the large weights problems in the binary treatment examples discussed by Kang et al. (2007), learning more accurate, bounded, and stable weights has been an active research area. Further techniques have proposed techniques to learn more robust propensity scores for binary treatments (Li et al., 2018; Zhao, 2019) too, however, the case of continuous treatments have received considerably less attention.

Entropy Balancing. To address the problem of extreme weights, *Entropy Balancing (EB)* (Hainmueller, 2012) estimates weights such that they balance the confounders subject to a measure of dispersion on the weights to prevent extremely large weights. Other loss functions using different dispersion metrics have been proposed for balancing (Zubizarreta, 2015; Chan et al., 2016). Zhao & Percival (2016) show that the entropy balancing is doubly robust. Entropy balancing has been extended to the continuous treatment setting (Fong et al., 2018; Vegetabile et al., 2021), where the balancing condition ensures that the weighted correlation between the confounders and the treatment is zero. Ai et al. (2021) propose a method for estimating the counterfactual distribution in the continuous treatment setting.

3. Methodology

To describe our end-to-end balancing algorithm, we first need to describe entropy balancing for continuous treatments with base weights.

3.1. Entropy Balancing for Continuous Treatments

Causal Inference via Entropy Balancing. Entropy balancing creates a pseudo-population using instance weights $w_i, i = 1, \ldots, n$, in which the treatment a and the confounders $\mathbf{x}$ are independent from each other. The independence is enforced by first selecting a set of functions on the confounders $\phi_k(\cdot) : \mathbb{X} \mapsto \mathbb{R}$, for $k = 1, \ldots, K$, that are dense and complete in L^2 space. Given the ϕ functions, we approximate the independence relationship by $\widehat{\mathbb{E}}_n[\mathrm{a}\phi_k(\mathbf{x})] = 0$, for $k = 1, \ldots, K$, where the empirical expectation $\widehat{\mathbb{E}}_n$ is performed on the pseudo population. Hereafter, we will denote the mapped data points as $\boldsymbol{\phi}(\boldsymbol{x}_i) = [\phi_1(\boldsymbol{x}_i), \ldots, \phi_K(\boldsymbol{x}_i)]$. The $\phi_k(\cdot)$ functions can be chosen based on prior knowledge or defined by the penultimate layer of a neural network that predicts (a, y) from $\mathbf{x}$. Our contributions in this paper are orthogonal to the choice of the $\phi_k(\cdot)$ functions and can benefit from ideas on learning these functions (Zeng et al., 2020). The data-driven choice of the number of bases K is beyond the scope of current paper and left to future work.

Balancing Constraint for the Continuous Treatments. Following (Fong et al., 2018; Vegetabile et al., 2021), in the case of continuous treatments, we first de-mean the confounders $\boldsymbol{\phi}(\boldsymbol{x}_i)$ and treatments a such that without loss of generality they are taken to have mean zero. The balancing objective is to learn a set of weights $w_i, i = 1, \ldots, n$ that satisfy $\sum_{i=1}^n w_i \boldsymbol{\phi}(\boldsymbol{x}_i) = \mathbf{0}$, $\sum_{i=1}^n w_i a_i = 0$, and $\sum_{i=1}^n w_i a_i \boldsymbol{\phi}(\boldsymbol{x}_i) = \mathbf{0}$. We can write these three constraints in a compact form by defining a $(2K+1)$–dimensional vector $\boldsymbol{g}_i = [\boldsymbol{\phi}(\boldsymbol{x}_i), a_i, a_i\boldsymbol{\phi}(\boldsymbol{x}_i)]$. The constraints become $\sum_{i=1}^n w_i \boldsymbol{g}_i = \mathbf{0}$. We stack the $\boldsymbol{g}$ vectors in a $(2K+1) \times n$ dimensional matrix $\boldsymbol{G}$ for compact notation. In this work, without loss of generality, we will present our idea with the first order balancing, without higher order moments (Galagate, 2016) or balancing in the kernel space (Wong & Chan, 2018; Kallus & Santacatterina, 2019; Hazlett, 2020).

Primal and Dual EB. A variety of dispersion metrics have been proposed as objective function for minimization such as entropy or variance of the weights (Wang & Zubizarreta, 2020). Hainmueller (2012) originally proposed minimizing the KL-divergence between the weights and a set of base weights $q_i, i = 1, \ldots, n$. Details on choice of base weights is discussed below, however, we note that $q_i = \text{const.}$ leads to minimization of the entropy of weights. Using this dispersion function and the balancing constraints, entropy balancing optimization is as follows:

$$\widehat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \sum_{i=1}^{n} w_i \log\left(\frac{w_i}{q_i}\right), \tag{1}$$

$$\begin{aligned} \text{s.t.} \quad & \text{(i) } \boldsymbol{G}\boldsymbol{w} = \mathbf{0}, \\ & \text{(ii) } \mathbf{1}^\top \boldsymbol{w} = 1, \\ & \text{(iii) } w_i \geq 0 \text{ for } i = 1, \ldots, n. \end{aligned}$$

The above optimization problem can be solved efficiently using its Lagrangian dual:

$$\widehat{\boldsymbol{\lambda}} = \underset{\boldsymbol{\lambda}}{\operatorname{argmin}} \; \log\left(\mathbf{1}^\top \exp\left(-\boldsymbol{\lambda}^\top \boldsymbol{G} + \boldsymbol{\ell}\right)\right), \tag{2}$$

where $\ell_i = \log q_i$ are the *log-base-weights*. Given the solution $\widehat{\boldsymbol{\lambda}}$, the balancing weights can be computed as $\boldsymbol{w} = \operatorname{softmax}\left(-\widehat{\boldsymbol{\lambda}}^\top \boldsymbol{G} + \boldsymbol{\ell}\right)$. The softmax function is defined as $\operatorname{softmax}(\boldsymbol{v}) = {}^{\exp \boldsymbol{v}}/\left(\mathbf{1}^\top \exp \boldsymbol{v}\right)$ for any vector $\boldsymbol{v}$. The log base weight is a degree of freedom that we have in the Eq. (2) to improve the quality of causal estimation. We select the mapping dimension K such that problem (2) is well-conditioned and leave the analysis of the high dimensional setting $K \approx n$ to future work. We can also add an L_1 penalty term to the dual objective in Eq. (2), which corresponds to approximate balancing (Wang & Zubizarreta, 2020).

In the next section we propose to parameterize the log-base-weights and learn them. Our analysis in Section 4 shows that with any arbitrary base weights, causal estimation using the weights learned in Eq. (2) will be consistent.

3.2. Learning the Base Weights

Hainmueller (2012) suggests two approaches for choosing base weights: (1) weights obtained from a conventional propensity score model and (2), in the context of survey design, using knowledge about the sampling design. We argue that a data-driven approach that learns customized base weights for each pair of dataset and weighted causal regression algorithm can further improve performance. For example, when we use weighted linear regression for causal inference, appropriate base weights can decrease the condition number of the design matrix and thus improve the quality of the regression. From another point of view, minimizing the KL-divergence between our weights and the base weights can act as a regularizer and improve the accuracy of weight estimation.

To address this problem, we define the log-base-weights $\boldsymbol{\ell}$ as a parametric function (e.g., a neural network) of the treatment variable; i.e. $\ell_{\boldsymbol{\theta}}(\cdot)$. We learn the base weights with the goal of improving the accuracy of the subsequent weighted regression. This is challenging because simply optimizing the weighted regression loss (e.g., weighted MSE) leads to degenerate results. That is, learning $\boldsymbol{\ell}$ to minimize the regression loss will lead to exclusion of the difficult-to-predict data points from the regression, which is undesirable. Thus, we need to find another loss function to optimize, ideally a loss function that directly minimizes the error in estimation of the response function $\mu(a)$.

Our idea for learning the parameters of the base weights is to generate multiple pseudo-responses $\overline{\mathrm{y}}$ with randomly generated response functions $\overline{\mu}(a)$. Now that we know the true response function $\overline{\mu}(a)$ in the randomly generated data, we can perform causal inference and obtain the estimation of the known response curve $\widehat{\mu}(a)$ using our weights. Algorithm 1 outlines our stochastic training of the base weight function. First, in Step 2, we estimate the distribution of noise using the residuals of regressing y over $(\mathrm{a}, \mathbf{x})$, capturing the possible heteroskedasticity in the noise. Then, in each iteration, we draw a batch of possible datasets. To generate each dataset, we randomly choose a response function $\overline{\mu}(a)$ and use it to generate the entire dataset (see Section 5.1 for examples of random functions). For the entire batch, we use $\ell_{\boldsymbol{\theta}}$ to learn the log-base-weights, and subsequently learn the weights in lines 7–8. In line 9 we use a weighted regression algorithm to find our estimation $\widehat{\mu}(a)$ of the randomly-generated $\overline{\mu}(a)$. Our loss function is the mean squared error between the latter quantities. While we call our algorithm *End-to-End Balancing* (E2B) because of our end-to-end optimization.

Algorithm 1 Stochastic Training of $\ell_{\boldsymbol{\theta}}$ for End-to-End Balancing

```
Require: Data tuples $(\boldsymbol{x}_i, a_i, y_i)$ for $i = 1, \ldots, n$ with an unknown response function $\mu(a)$.
Require: Representation functions $\boldsymbol{\phi}(\cdot)$ and $\boldsymbol{\psi}(\cdot)$, split size $n_1 < n$ and batch size $B$.
Generate a random set of indexes $I, |I| = n_1$ and its complement $I^c$ and split the data to $S$ and $S^c$ using them.
Estimate the distribution of noise in $\mathrm{y}$ given $(\mathrm{a}, \mathbf{x})$ as $\widehat{F}_{\varepsilon}$.
Compute $\boldsymbol{G}$ by stacking $\boldsymbol{g}_i = [\boldsymbol{\phi}(\boldsymbol{x}_i), a_i, a_i\boldsymbol{\phi}(\boldsymbol{x}_i)]$, for $i = 1, \ldots, n$.
for Number of Iterations do
    Generate $B$ datasets $\{(\boldsymbol{x}_i, a_i, \overline{y}_{i,b})\}_{i=1}^{n}$ for $b = 1, \ldots, B$ using $\varepsilon \sim \widehat{F}_{\varepsilon}$, and randomly selected $\overline{\mu}(a)_b$ response functions.
    $\ell_i \leftarrow \ell_{\boldsymbol{\theta}}(\boldsymbol{\psi}(a_i, \boldsymbol{x}_i))$.
    $\widehat{\boldsymbol{\lambda}} \leftarrow \mathrm{argmin}_{\boldsymbol{\lambda}} \left\{\log\left(\mathbf{1}^\top \exp\left(-\boldsymbol{\lambda}^\top \boldsymbol{G} + \boldsymbol{\ell}\right)\right) + \gamma\|\boldsymbol{\lambda}\|_1\right\}$ using only $S$ data.
    $\boldsymbol{w} \leftarrow \mathrm{softmax}\left(-\widehat{\boldsymbol{\lambda}}^\top \boldsymbol{G} + \boldsymbol{\ell}\right)$ using only $S^c$ data.
    $\widehat{\mu}(a)_b \leftarrow$ weighting-based causal estimates using $(a_i, \overline{y}_{i,b}, w_i)$ in $S^c$ for $b = 1, \ldots, B$.
    Take a step in $\boldsymbol{\theta}$ to minimize $\frac{1}{B}\sum_{b=1}^{B}\left(\widehat{\mu}(a)_b - \overline{\mu}(a)_b\right)^2$.
end for
return The $\ell_{\widehat{\boldsymbol{\theta}}}$ function.
```

Sample Splitting. The E2B procedure involves estimation of two sets of parameters $\boldsymbol{\theta}$ in the $\ell_{\boldsymbol{\theta}}$ and $\boldsymbol{\lambda}$ for entropy balancing. The joint estimation of $\boldsymbol{\theta}, \boldsymbol{\lambda}$ on a single sample will result in bias (Chernozhukov et al., 2018). Thus, we split the sample to two mutually exclusive parts and perform the optimizations on separate partitions of data.

Choice of Random Response Functions. Ideally, we should rely on domain experts for choosing the random set of response functions $\overline{\mu}(a)$ that includes the true response function. Alternatively, we can choose broad function classes such as random piecewise smooth functions or polynomial functions with random coefficients. We can also use generative adversarial networks to generate data that is more similar to our sample (Athey et al., 2021).

Features Fed to $\ell_{\boldsymbol{\theta}}$. We can feed the raw values of the treatments and any handcrafted features, denoted by $\boldsymbol{\psi}(a_i, \boldsymbol{x}_i)$. We empirically find that $\boldsymbol{\psi}(a_i, \boldsymbol{x}_i) = (\log p(a_i), \log p(a_i|\boldsymbol{x}_i))$ makes training the $\ell_{\boldsymbol{\theta}}$ easier. These features are the logarithms of the nominator and denominator of the stable weights. Given this choice, we can visualize the $\ell_{\boldsymbol{\theta}}$ function and find its relationship to the marginal and conditional distributions of the treatment. We describe the details of our neural network model for $\ell_{\boldsymbol{\theta}}$ and our techniques for training in Appendix B.

Weighted Regression Algorithms. To be able to differentiate the loss function with respect to $\boldsymbol{\theta}$, we need weighted regression algorithms whose estimates are differentiable with respect to the weights. In the linear average treatment effect function we choose weighted linear regression and in the non-linear setting we use the weighted polynomial regression and the local kernel regression, as used by Flores et al. (2012).

Double Robustness. Zhao & Percival (2016) show that in the binary case, the entropy balancing is doubly robust. We do not attempt to show double robustness for E2B because we see E2B as a meta algorithm that learns customized weights for each dataset and algorithm. We can either (1) plug-in the E2B weights in the doubly robust algorithm and expect improved accuracy, or (2) learn weights that directly minimize the error of doubly robust algorithms such as (Díaz & van der Laan, 2013; Kennedy et al., 2017). However, if we use an outcome regression model for generating the random response functions $\overline{\mu}(a)$, the E2B weights may no longer provide significant improvements to the doubly robust techniques.

4. Analysis

We prove that for any arbitrary choice of the log-base-weight function $\ell_{\boldsymbol{\theta}}$, our approach consistently estimates causal effects. Before proving the consistency results, we characterize the quantity that our solution converges to. All long proofs are relegated to Appendix A.

Definition 1. ***Generalized Stable Weights.*** *Suppose $f(a, \boldsymbol{x})$ denote the joint probability density function of treatments and confounders in a population. Suppose $\widetilde{f}(a)$ and $\widetilde{f}(\boldsymbol{x})$ denote two arbitrary density functions, possibly different with the marginal density functions in our population, that satisfy $\mathbb{E}_{\mathbf{x}\sim\widetilde{f}(\boldsymbol{x})}[\mathbf{x}] = \mathbf{0}$ and $\mathbb{E}_{\mathrm{a}\sim\widetilde{f}(a)}[\mathrm{a}] = 0$. We define the Generalized Stable Weights as follows*

$$w_{GSW}(a, \boldsymbol{x}) = \frac{\widetilde{f}(a)\widetilde{f}(\boldsymbol{x})}{f(a, \boldsymbol{x})}. \tag{3}$$

Remark. Our definition generalizes the stabilized weights defined by Robins et al. (2000), where $\widetilde{f}(a)$ and $\widetilde{f}(\boldsymbol{x})$ match the marginal probability density functions in the original population.

Proposition 1. *The generalized stable weights w_{GSW} satisfy $\mathbb{E}\left[w_{GSW}\mathrm{a}\mathbf{x}\right] = \mathbf{0}$.*

Proof.

$$\begin{aligned}\mathbb{E}\left[\mathrm{w}_{GSW}\mathrm{a}\mathbf{x}\right] &= \mathbb{E}\left[\frac{\widetilde{f}(\mathrm{a})\widetilde{f}(\mathbf{x})}{f(\mathrm{a},\mathbf{x})}\mathrm{a}\mathbf{x}\right]\\ &= \int\int\frac{\widetilde{f}(a)\widetilde{f}(\boldsymbol{x})}{f(a,\boldsymbol{x})}a\boldsymbol{x}\mathrm{d}F_{\mathrm{a},\mathbf{x}}(a,\boldsymbol{x})\\ &= \int a\widetilde{f}(a)\mathrm{d}a\int\boldsymbol{x}\widetilde{f}(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \mathbf{0},\end{aligned}$$

where the last equation is because of zero mean assumption for the $\widetilde{f}(a)$ and $\widetilde{f}(\boldsymbol{x})$ distributions. □

Now, we can show that with an appropriate choice of the $\boldsymbol{\phi}$ functions, the solution of Eq. (2) approximates the generalized stable weights. Consider the population version of Eq. (2):

$$\boldsymbol{\lambda}^\star = \operatorname*{argmin}_{\boldsymbol{\lambda}} \log\left(\mathbb{E}\left[\exp(\mathbf{g}^\top\boldsymbol{\lambda}+\ell)\right]\right). \tag{4}$$

The weights corresponding to $\boldsymbol{\lambda}^\star$ can be calculated as $w^\star = C\exp(\mathbf{g}^\top\boldsymbol{\lambda}^\star+\ell)$, where $C = \left(\int\exp(\mathbf{g}^\top\boldsymbol{\lambda}^\star+\ell)\mathrm{d}F(a,\boldsymbol{x})\right)^{-1}$ is the normalization constant.

Assumptions.

1. $f(a,\boldsymbol{x}) \geq c > 0$ for all $(a,\boldsymbol{x}) \in \mathbb{A}\times\mathbb{X}$ pairs, where c is a constant.
2. Suppose the basis functions are dense and rich enough such for some small values of $\delta_{\boldsymbol{\phi}_K}$ that they satisfy:

   $$\mathbb{E}[\mathrm{a}\boldsymbol{\phi}(\mathbf{x})] = \mathbf{0} \text{ only if } \sup_{a,\boldsymbol{x}}|f(a,\boldsymbol{x}) - f(a)f(\boldsymbol{x})| = \delta_{\boldsymbol{\phi}_K}.$$
3. Suppose the population problem in Eq. (4) has a unique solution $\boldsymbol{\lambda}^\star$ and the corresponding weights are denoted by $w^\star$.

The following theorem shows that the solution to Eq. (4) converges to w_{GSW}:

Theorem 1. *Given the assumptions, the solution to the population problem satisfies:*

$$\sup_{a,\boldsymbol{x}}|w^\star(a,\boldsymbol{x}) - w_{GSW}(a,\boldsymbol{x})| \leq \delta_{\boldsymbol{\phi}_K}/c. \tag{5}$$

If we select the function set $\boldsymbol{\phi}_K$ such that $\delta_{\boldsymbol{\phi}_K} = o(1)$, the theorem shows that $w^\star(a,\boldsymbol{x})$ is an unbiased estimator of $w_{GSW}(a,\boldsymbol{x})$. Notice that Assumption 1 is only slightly stronger than the common positivity assumption. Assumption 2 requires us to select the mapping functions such that zero the correlation between the mapped confounders and the treatment implies their independence. We provide the proof in Appendix A.1.

Note that the quality of the $\boldsymbol{\psi}$ features and neural network training does not affect the unbiasedness of the E2B because of the balancing constraint is still satisfied. The flexibility in choice of $\widetilde{f}$ distributions in the definition of w_{GSW} is due to the fact that we require only first order balancing. If we enforce higher order balancing constraints in the form of $\mathbb{E}[\mathrm{w}^\star\phi_1(\mathrm{a})\phi_2(\mathbf{x})] = \mathbb{E}[\phi_1(\mathrm{a})]\cdot\mathbb{E}[\phi_2(\mathbf{x})]$ for any suitable functions ϕ_1 and ϕ_2, Theorem 1 in (Ai et al., 2021) shows that $w^\star = {f(a)}/{f(a|\boldsymbol{x})}$. The more flexible form of weights in Eq. (3) allows us to pick the marginals $\widetilde{f}(a)$ and $\widetilde{f}(\boldsymbol{x})$ with more freedom. In this work, we have chosen a data-driven way to learn them.

Finally, the following theorem establishes the asymptotic consistency and normality result for each individual weight estimated by E2B, under the common regularity conditions for problem (2).

Theorem 2. *Suppose $\Lambda \subset \mathbb{R}^{2K+1}$ is an open subset of Euclidean space and the solution $\widehat{\boldsymbol{\lambda}}_n \in \Lambda$ to Eq. (2) is within the subset. The weights estimated by Eq. (2) are asymptotically normal for $i = 1,\ldots,n$:*

$$\sqrt{n}\left(\,\widehat{w}_n(a_i,\boldsymbol{x}_i) - w^\star(a_i,\boldsymbol{x}_i)\,\right) \xrightarrow{d} \mathcal{N}(0,\sigma^2(a_i,\boldsymbol{x}_i)). \tag{6}$$

We provide the population form of $\sigma^2(a_i,\boldsymbol{x}_i)$ and an unbiased sample estimate for it in Appendix A.2.

5. Experiments

We use two synthetic and one real-world datasets to show that E2B outperforms the baselines. In the synthetic datasets, we have access to the true treatment effects; thus we measure accuracy of the algorithms in recovering the treatment effects. In the real-world data, we qualitatively evaluate the estimated causal treatment effect curve and inspect the learned log-base-weight function.

Baselines. A key baseline in our study is the Inverse Propensity score Weighting (IPW) with Stable Weights (Robins et al., 2000) as the most commonly used technique. To avoid extreme weights and prevent instability, we trim (Winsorize) the weights by $[5, 95]$ percentiles (Cole & Hernán, 2008; Crump et al., 2009; Chernozhukov et al., 2018). However, the main baseline in our experiments is Entropy Balancing (Vegetabile et al., 2021), which is equal to E2B with $\ell_{\boldsymbol{\theta}} = \text{const}$, corresponding to the constant base weights. EB allows us to do an ablation study and see the exact amount of improvement by learning a customized $\ell_{\boldsymbol{\theta}}$ function. We also include EB with the stabilized weights (SW) as base weights ($\ell_{\boldsymbol{\theta}} = \log\widehat{p}(a) - \log\widehat{p}(a|\boldsymbol{x})$). For fair comparison to the entropy balancing methods, we only include first order balancing methods, as our idea of learning the base-weights can be combined with the higher order and kernel-based balancing methods. Finally, we also include the permutation weighting algorithm (Arbour et al., 2021) that proposes to compute the weights using permutation of the treatments and a classifier that predict the probability of being permuted. We provide further details on this algorithm in Appendix B.4.

Training Details. We provide the details of the neural networks used for the $\ell_{\boldsymbol{\theta}}$ and propensity score estimation for IPW in Appendix B. All neural networks are trained using Adam (Kingma & Ba, 2014) with early stopping based on validation error. The learning rate and architectural parameters of the neural networks are tuned via hyperparameter search on the validation data.

5.1. Synthetic Data Experiments

Linear. We use the following steps to generate 100 datasets, each with 1000 data points.

1. Generate confounders $\mathbf{x} \in \mathbb{R}^5$, $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a tridiagonal covariance matrix with diagonal and off-diagonal elements equal to 1.0 and 0.2, respectively.

2. $\mathrm{a} \sim \mathcal{N}(\mu_a, 0.3^2)$, where $\mu_a = \sin(\boldsymbol{\beta}_{xa}^\top \mathbf{x})$ and $\beta_{xa,k} \sim \mathrm{Unif}(-1, 1)$ for $k = 1, \ldots, 5$.

3. $\mathrm{y} \sim \mathcal{N}(\mu_y, 0.5^2)$, where $\mu_y = \boldsymbol{\beta}_{xy}^\top \mathbf{x} + \beta_{ay}\mathrm{a}$, where $\beta_{ax}, \beta_{xy,k} \sim \mathcal{N}(0, 1)$ for $k = 1, \ldots, 5$.

We use weighted least squares as the regression algorithm and report the average $|\widehat{\beta_{ay}} - \beta_{ay}|$ over all 100 datasets.

Nonlinear We first generate confounders $\mathbf{x}$ and treatments a similar to steps 1 and 2 of the linear case. Then, we generate the response variable according to $\mathrm{y} \sim \mathcal{N}(\mu_y, 0.5^2)$, where $\mu_y = \boldsymbol{\beta}_{xy}^\top \mathbf{x} + h_{\boldsymbol{\gamma}_{ay}}(\mathrm{a})$, where $\beta_{xy,k} \sim \mathcal{N}(0, 1)$ for $k = 1, \ldots, 5$. The Hermit polynomials are defined as $h_{\boldsymbol{\gamma}}(z) = \gamma_0 + \gamma_1 z + \gamma_2(z^2 - 1) + \gamma_3(x^3 - 3x)$. Similar to the linear case, we generate 100 samples of size 1000. We use the weighted polynomial regression as the regression algorithm to estimate $\widehat{\boldsymbol{\gamma}}$ and report the average RMSE between true $\boldsymbol{\gamma}$ and $\widehat{\boldsymbol{\gamma}}$. We report the mean and standard error of errors on 100 datasets in Table 1.

As seen in Table 1, in both linear and non-linear datasets, the E2B is significantly more accurate in uncovering the true treatment response functions. Both constant and IPW base weights perform worse than the base-weights learned by end-to-end balancing. As Robins et al. (2007) caution, synthetic data evaluation might exacerbate the extreme weights issues because unlike real data, usually no manual inspection of weights are done.

To gain more insights, in Figure 1, we plot the log-base weight function that we learn as a function of $\log(\widehat{p}(a))$ and $\log(\widehat{p}(a|\mathbf{x}))$. We align all curves at their starting point and plot the median of 100 runs. Both figures, show more variations in the $\log(\widehat{p}(a))$–axis, rather than the $\log(\widehat{p}(a|\mathbf{x}))$–axis. Not that, especially in the linear case, the smaller conditional probability leads to larger base-weights, inline with the IPW base-weights. Finally, the complexity of the plots emphasizes the need for end-to-end methods for learning weights. Given the results in Figure 1, the learned log-base-weight does not seem to be a convex combination of uniform and SW base weight.

5.2. Real Data Experiments

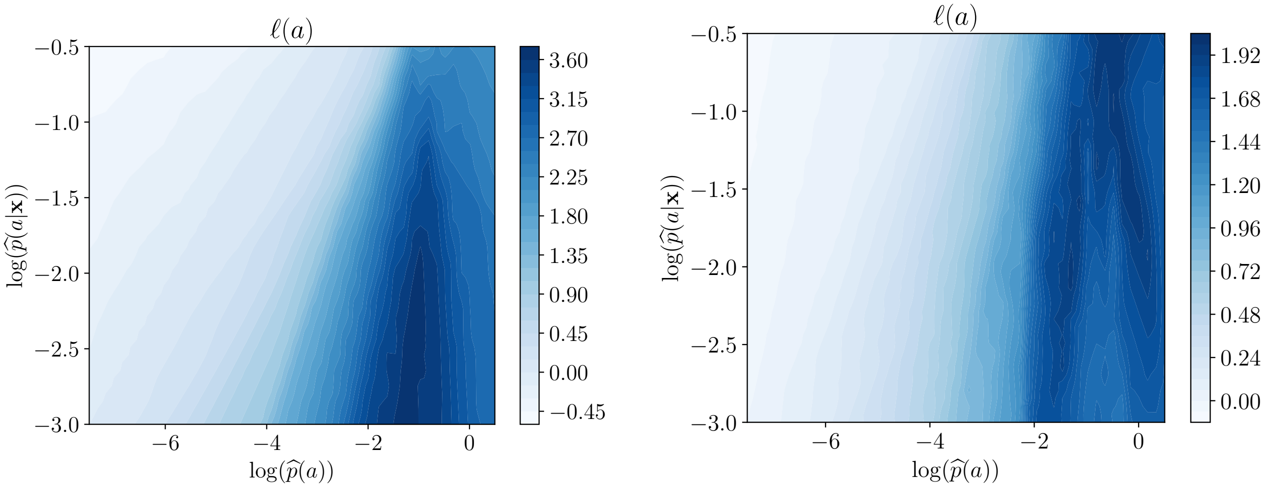
We study the impact of $PM_{2.5}$ particle level on the cardiovascular mortality rate (CMR) in 2132 counties in the US using the data provided by the National Studies on Air Pollution and Health (Rappold, 2020). The data is publicly available under U.S. Public Domain license. The $PM_{2.5}$ particle level and the mortality rate are measured by $\mu g/m^3$ and the number of annual deaths due to cardiovascular conditions per 100,000 people, respectively. We use only the data for 2010 to simplify the experiment setup; thus we measure the same year impact of $PM_{2.5}$ particle level. Other than the treatment and response variables, the data includes 10 variables such as poverty rate, population, and household income, which we use as confounders. We provide the descriptive statistics and the histograms for the treatment and effect in Appendix C.

To train E2B, we create the random dataset (Line 5 in Algorithm 1) using Hermite polynomials of max degree 3, $\mu_y = \left|h_{\boldsymbol{\gamma}_{xy}}(\boldsymbol{\beta}_{xy}^\top \mathbf{x} / \|\boldsymbol{\beta}_{xy}^\top \mathbf{x}\|_2) + h_{\boldsymbol{\gamma}_{ay}}(\mathrm{a})\right|$. We use absolute value to capture the positivity of our response variable. The data also shows heteroskedasticity; we model the noise as a zero mean Gaussian variable with variance $\sigma^2(\widehat{y}) = 6.00\widehat{y}$. For regression, we use the non-parametric local kernel regression algorithm. We measure the uncertainty in the curves using the deep ensembles technique (Lakshminarayanan et al., 2017) with 100 random ensembles. That means, in each experiment, we initialize the neural network with different random values. To further improve the uncertainty estimation, in each training, we resample the dataset too.

Figure 2a shows the average treatment effect curve for the impact of $PM_{2.5}$ on CMR. We show the one standard deviation interval using the shaded areas. Starting around $PM_{2.5} = 5.3\mu g/m^3$ the curve increases with a steep slope; confirming the previous studies that increased $PM_{2.5}$ levels increase the probability of cardiovascular mortality. Our results are generally aligned with the results reported in (Wu et al., 2020). We can see that after $PM_{2.5} = 6.4\mu g/m^3$ the curve plateaus and mortality rate stays at elevated lev-

Table 1: Average RMSE for estimation of the response functions. The results are in the form of "mean (std. err.)" from 100 runs.

Algorithm	Linear	Non-linear
Inverse Propensity Weighting (SW)	2.057 (0.437)	0.530 (0.025)
Permutation Weighting	1.1543 (6.580)	0.525 (0.250)
Entropy Balancing (Const.)	0.880 (0.072)	0.335 (0.022)
Entropy Balancing (SW)	0.652 (0.059)	0.403 (0.025)
End-to-End Balancing	$\mathbf{0.383}$ $(\mathbf{0.035})$	$\mathbf{0.276}$ $(\mathbf{0.014})$



(a) Linear Design

(b) Non-linear Design

Figure 1: The estimated log-base-weight function $\ell_{\boldsymbol{\theta}}$ as a function of logarithms of the empirical density of the treatment $\log(\widehat{p}(a))$ and conditional distribution $\log(\widehat{p}(a|\mathbf{x}))$. We perform the experiment 100 times and report the median and the inter-quantile range. We align all curves by normalizing their value at the beginning to zero.

els. Looking at the histogram of the treatments in Figure 3a in the appendix, we observed that most counties have $PM_{2.5}$ between 6 and 8. This might justify the fluctuations that we see in this interval and may allude about potential unmeasured confounders.

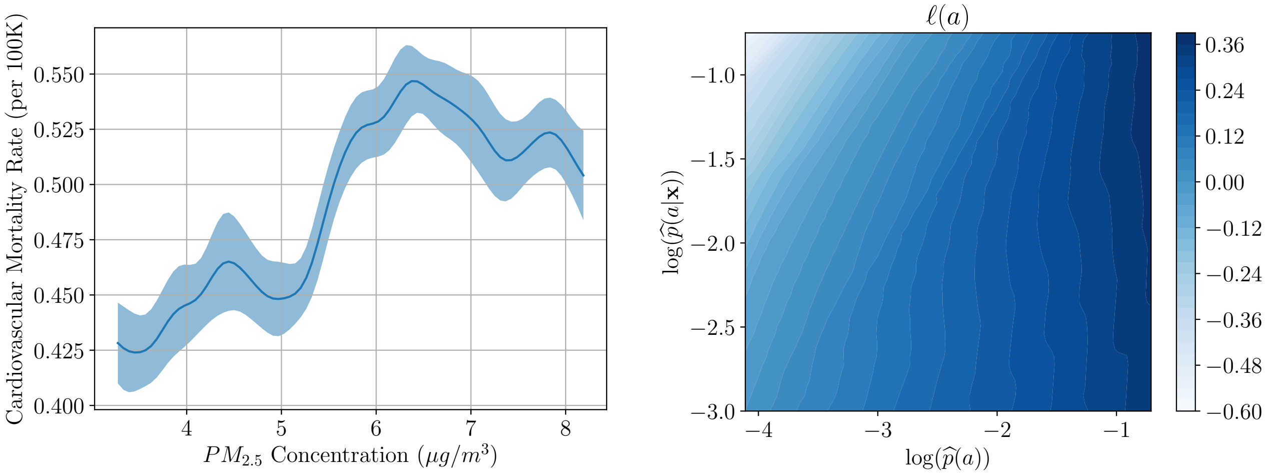
Figure 2b shows the log-base-weight function that we learn in this data. Similar to the synthetic experiments, we show the median of 100 runs. While the plot shows smaller variations, it is generally inline with the observations we had in the synthetic data.

6. Discussion

Causal inference is a well-studied problem; its main goal is to remove biases due to confounding by balancing the population to look similar to randomized controlled trials. Removing the impact of confounders can play a critical role in reducing and possibly eliminating bias in our decision making leading to potentially positive societal impacts. Our results rely on two classical assumptions: (1) unconfoundedness and (2) positivity. While these assumptions are sometimes reasonable in practice, their violations might lead to biased causal inferences. For example, the positivity assumption might be violated if we do not collect any data for a sub-population. Overall, the debiasing property of causal inference should not relieve us from rigorous data collection and analysis setup. In our experiments, we have been careful to quantify uncertainty in our causal estimation and be wary of over-confidence in our results. We performed our experiments on a CPU machine with 16 cores from a cloud provider that uses hydroelectric power.

7. Conclusion

We observed that in the entropy balancing framework, the base weights provide an extra degree of freedom to optimize the accuracy of causal inference. We propose end-to-end



(a) Response Curve

(b) Log-base-weights function

Figure 2: (a) The average treatment effect curve for measuring the impact of $PM_{2.5}$ concentration on the cardiovascular mortality rate. We perform the experiment 100 times and report the mean and ±std range. (b) The estimated log-base-weight function $\ell_{\boldsymbol{\theta}}$ as a function of logarithm of the empirical density of the treatment $\log(\widehat{p}(a))$.

balancing (E2B) as a technique to learn the base weight such that they directly improve the accuracy of causal inference using end-to-end optimization. In our theoretical analysis we find the quantity that E2B weights are approximating and discuss E2B's statistical consistency. Using synthetic and real-world data, we show that our proposed algorithm outperforms the entropy balancing in terms of causal inference accuracy.

References

Ai, C., Linton, O., and Zhang, Z. Estimation and inference for the counterfactual distribution and quantile functions in continuous treatment models. *Journal of Econometrics*, 2021.

Arbour, D., Dimmery, D., and Sondhi, A. Permutation weighting. In *ICML*, 18–24 Jul 2021.

Athey, S., Imbens, G. W., Metzger, J., and Munro, E. Using wasserstein generative adversarial networks for the design of monte carlo simulations. *Journal of Econometrics*, 2021.

Chan, K. C. G., Yam, S. C. P., and Zhang, Z. Globally efficient non-parametric inference of average treatment effects by empirical balancing calibration weighting. *Journal of the Royal Statistical Society. Series B, Statistical methodology*, 78(3):673, 2016.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W., and Robins, J. Double/debiased machine learning for treatment and structural parameters: Double/debiased machine learning. *The Econometrics Journal*, 21(1), 2018.

Cole, S. R. and Hernán, M. A. Constructing inverse probability weights for marginal structural models. *American journal of epidemiology*, 168(6):656–664, 2008.

Crump, R. K., Hotz, V. J., Imbens, G. W., and Mitnik, O. A. Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96(1):187–199, 2009.

Díaz, I. and van der Laan, M. J. Targeted data adaptive estimation of the causal dose–response curve. *Journal of Causal Inference*, 1(2):171–192, 2013.

Flores, C. A., Flores-Lagunes, A., Gonzalez, A., and Neumann, T. C. Estimating the effects of length of exposure to instruction in a training program: the case of job corps. *Review of Economics and Statistics*, 94(1): 153–171, 2012.

Fong, C., Hazlett, C., Imai, K., et al. Covariate balancing propensity score for a continuous treatment: application to the efficacy of political advertisements. *The Annals of Applied Statistics*, 12(1):156–177, 2018.

Galagate, D. *Causal Inference With a Continuous Treatment and Outcome: Alternative Estimators for Parametric Dose-response Functions With Applications*. PhD thesis, University of Maryland, 2016.

Hainmueller, J. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political analysis*, pp. 25–46, 2012.

Hazlett, C. Kernel balancing: A flexible non-parametric weighting procedure for estimating causal effects. *Statistica Sinica*, 2020.

Hirano, K. and Imbens, G. W. The propensity score with continuous treatments. *Applied Bayesian modeling and causal inference from incomplete-data perspectives*, 226164:73–84, 2004.

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

Imbens, G. W. and Rubin, D. B. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

Kallus, N. and Santacatterina, M. Kernel optimal orthogonality weighting: A balancing approach to estimating effects of continuous treatments. *arXiv preprint arXiv:1910.11972*, 2019.

Kang, J. D., Schafer, J. L., et al. Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical science*, 22(4):523–539, 2007.

Kennedy, E. H., Ma, Z., McHugh, M. D., and Small, D. S. Nonparametric methods for doubly robust estimation of continuous treatment effects. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, 79(4): 1229, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, pp. 6405–6416, 2017.

Li, F., Morgan, K. L., and Zaslavsky, A. M. Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113(521):390–400, 2018.

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

Peters, J., Janzing, D., and Schölkopf, B. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

Rappold, A. Annual PM2.5 and cardiovascular mortality rate data: Trends modified by county socioeconomic status in 2, 132 US counties, 2020. URL https://edg.epa.gov/metadata/catalog/search/resource/details.page?uuid=https://doi.org/10.23719/1506014.

Robins, J., Hernán, M., and Brumback, B. Marginal structural models and causal inference in epidemiology. *Epidemiology*, 11(5):550–560, 2000.

Robins, J., Sued, M., Lei-Gomez, Q., and Rotnitzky, A. Comment: Performance of double-robust estimators when" inverse probability" weights are highly variable. *Statistical Science*, 22(4):544–559, 2007.

Smith, J. A. and Todd, P. E. Does matching overcome lalonde's critique of nonexperimental estimators? *Journal of econometrics*, 125(1-2):305–353, 2005.

Spirtes, P., Glymour, C. N., Scheines, R., and Heckerman, D. *Causation, prediction, and search*. MIT press, 2000.

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

Vegetabile, B. G., Griffin, B. A., Coffman, D. L., Cefalu, M., Robbins, M. W., and McCaffrey, D. F. Nonparametric estimation of population average dose-response curves using entropy balancing weights for continuous exposures. *Health Services and Outcomes Research Methodology*, 21(1):69–110, 2021.

Wang, Y. and Zubizarreta, J. R. Minimal dispersion approximately balancing weights: asymptotic properties and practical considerations. *Biometrika*, 107(1):93–105, 2020.

Wong, R. K. and Chan, K. C. G. Kernel-based covariate functional balancing for observational studies. *Biometrika*, 105(1):199–213, 2018.

Wu, X., Braun, D., Schwartz, J., Kioumourtzoglou, M., and Dominici, F. Evaluating the impact of long-term exposure to fine particulate matter on mortality among the elderly. *Science advances*, 6(29):eaba5692, 2020.

Zeng, S., Assaad, S., Tao, C., Datta, S., Carin, L., and Li, F. Double robust representation learning for counterfactual prediction. *arXiv:2010.07866*, 2020.

Zhao, Q. Covariate balancing propensity score by tailored loss functions. *The Annals of Statistics*, 47(2):965–993, 2019.

Zhao, Q. and Percival, D. Entropy balancing is doubly robust. *Journal of Causal Inference*, 5(1), 2016.

Zhu, Y., Coffman, D. L., and Ghosh, D. A boosting algorithm for estimating generalized propensity scores with continuous treatments. *Journal of causal inference*, 3(1): 25–40, 2015.

Zubizarreta, J. R. Stable weights that balance covariates for estimation with incomplete outcome data. *Journal of the American Statistical Association*, 110(511):910–922, 2015.

Zubizarreta, J. R., Reinke, C. E., Kelz, R. R., Silber, J. H., and Rosenbaum, P. R. Matching for several sparse nominal variables in a case-control study of readmission following surgery. *The American Statistician*, 65(4):229–238, 2011.

A. Proofs of the Theorems

Whenever the context of an expectation operation is not clear, we disambiguate it by specifying the variable that the expectation is taken over and its distribution $\mathbb{E}_{\mathbf{x}\sim f(\boldsymbol{x})}[\mathbf{x}]$.

A.1. Proof of Theorem 1

Proof. Given that the logarithm function is a strictly increasing function, we can omit it in the optimization; i.e., $\boldsymbol{\lambda}^\star = \operatorname{argmin}_{\boldsymbol{\lambda}} \mathbb{E}\left[\exp(\mathbf{g}^\top\boldsymbol{\lambda}+\ell)\right]$. Because this is an unconstrained optimization, the optimal solution occurs when the gradient is equal to zero.

$$\begin{aligned}\mathbb{E}[\mathbf{g}\exp\left(\mathbf{g}^\top\boldsymbol{\lambda}^\star+\ell\right)] &= \mathbf{0},\\ \mathbb{E}[\mathbf{g}w^\star(\mathrm{a},\mathbf{x})] &= \mathbf{0},\end{aligned} \tag{7}$$

where the last equation is due to the equation of the weights in the population optimization problem.

Using the definition for the $\mathbf{g}$ vector, Eq. (7) implies that $\mathbb{E}[w^\star(\mathrm{a},\mathbf{x})\mathrm{a}\boldsymbol{\phi}(\mathbf{x})] = \mathbf{0}$. Thus, we conclude that in the weighted population (with distribution $\widetilde{F}$), the a and $\boldsymbol{\phi}(\mathbf{x})$ are uncorrelated:

$$\mathbb{E}_{(\mathrm{a},\mathbf{x})\sim\widetilde{F}}[\mathrm{a}\boldsymbol{\phi}(\mathbf{x})] = \mathbf{0} \tag{8}$$

For every set $\mathcal{B}\subset\mathbb{A}\times\mathbb{X}$, we can write:

$$\widetilde{F}(\mathcal{B}) = \int_{\mathcal{B}} w^\star(a,\boldsymbol{x})\mathrm{d}F(a,\boldsymbol{x}). \tag{9}$$

The Radon-Nikodym theorem implies that $w^\star(a,\boldsymbol{x})$ is the Radon-Nikodym derivative:

$$\begin{aligned} w^\star(\boldsymbol{x},a) &= \frac{\mathrm{d}\widetilde{F}(\boldsymbol{x},a)}{\mathrm{d}F(\boldsymbol{x},a)} = \frac{\widetilde{f}(\boldsymbol{x},a)}{f(\boldsymbol{x},a)} \qquad (10)\\ &= \frac{\widetilde{f}(\boldsymbol{x})\widetilde{f}(a)+\Big\{\widetilde{f}(\boldsymbol{x},a)-\widetilde{f}(\boldsymbol{x})\widetilde{f}(a)\Big\}}{f(\boldsymbol{x},a)}, \qquad (11)\\ &= w_{GSW}(a,\boldsymbol{x}) + \frac{\widetilde{f}(\boldsymbol{x},a)-\widetilde{f}(\boldsymbol{x})\widetilde{f}(a)}{f(\boldsymbol{x},a)} \qquad (12)\end{aligned}$$

Thus, using Eq. (8) and Assumptions 1 and 3 we can write

$$\sup_{a,\boldsymbol{x}} |w^\star(a,\boldsymbol{x}) - w_{GSW}(a,\boldsymbol{x})| \le \delta_{\boldsymbol{\phi}_K}/c.$$

□

A.2. Theorem 2

Proof. Given that the logarithm function is a strictly increasing function, we can omit it in the optimizations. Thus the sample and population solutions are:

$$\begin{aligned}\widehat{\boldsymbol{\lambda}}_n &= \operatorname*{argmin}_{\boldsymbol{\lambda}} \frac{1}{n}\sum_{i=1}^{n}\exp(\boldsymbol{g}_i^\top\boldsymbol{\lambda}+\ell_i),\\ \boldsymbol{\lambda}^\star &= \operatorname*{argmin}_{\boldsymbol{\lambda}} \mathbb{E}\left[\exp(\mathbf{g}^\top\boldsymbol{\lambda}+\ell)\right].\end{aligned}$$

The estimator is an M-estimator and given our sample-splitting, the proof follows the asymptotic normality of the estimator (Van der Vaart, 2000, Chapter 5.3).

$$\sqrt{n}\left(\widehat{\boldsymbol{\lambda}}_n - \boldsymbol{\lambda}^\star\right) \overset{d}{\to} \mathcal{N}(\mathbf{0},\boldsymbol{V}), \tag{13}$$

To obtain the value of $\boldsymbol{V}_{\mathbf{1}}$, note that the optimal sample solution occurs at the solution of the following equation (Z-estimator equation):

$$\sum_{i=1}^{n}\boldsymbol{g}_i\exp\left(\boldsymbol{g}_i^\top\widehat{\boldsymbol{\lambda}}_n\right) = \mathbf{0}.$$

Thus, the score function is $\boldsymbol{\psi}_{\boldsymbol{\lambda}} = \boldsymbol{g}_i \exp\left(\boldsymbol{g}_i^\top \boldsymbol{\lambda}\right)$. We denote the matrix of derivatives of the score function by $\dot{\boldsymbol{\psi}}_{\boldsymbol{\lambda}}$ whose elements are defined as $\dot{\psi}_{\boldsymbol{\lambda},kk'} = {\partial \psi_{\boldsymbol{\lambda},k}}/{\partial \lambda_{k'}}$. Using the theorem in (Van der Vaart, 2000, Chapter 5.3), we can write:

$$\boldsymbol{V} = \mathbb{E}[\dot{\boldsymbol{\psi}}_{\boldsymbol{\lambda}^\star}]^{-1}\mathbb{E}[\boldsymbol{\psi}_{\boldsymbol{\lambda}^\star}\boldsymbol{\psi}_{\boldsymbol{\lambda}^\star}^\top]\mathbb{E}[\dot{\boldsymbol{\psi}}_{\boldsymbol{\lambda}^\star}]^{-1}. \tag{14}$$

In the above equation we have assumed that $\mathbb{E}[\dot{\boldsymbol{\psi}}_{\boldsymbol{\lambda}^\star}]$ matrix is invertable. An unbiased sample estimation of $\boldsymbol{V}$ can be obtained by substituting $\widehat{\boldsymbol{\lambda}}_n$ in place of $\boldsymbol{\lambda}^\star$ and taking empirical expectations.

An application of the delta method on Eq. (13) yields:

$$\sqrt{n}\left(\frac{\exp\left(\boldsymbol{g}_i^\top \widehat{\boldsymbol{\lambda}}_n + \ell_i\right)}{\frac{1}{n}\sum_{i=1}^n \exp\left(\boldsymbol{g}_i^\top \widehat{\boldsymbol{\lambda}}_n + \ell_i\right)} - \frac{\exp\left(\boldsymbol{g}_i^\top \boldsymbol{\lambda}^\star + \ell_i\right)}{\frac{1}{n}\sum_{i=1}^n \exp\left(\boldsymbol{g}_i^\top \boldsymbol{\lambda}^\star + \ell_i\right)}\right) \overset{d}{\to} \mathcal{N}(\mathbf{0}, \sigma^2), \tag{15}$$

$$\sqrt{n}\left(\widehat{w}_n(a_i, \boldsymbol{x}_i) - \frac{\exp\left(\boldsymbol{g}_i^\top \boldsymbol{\lambda}^\star + \ell_i\right)}{\mathbb{E}[\exp(\mathbf{g}^\top \boldsymbol{\lambda}^\star + \lambda)]}\right) \overset{d}{\to} \mathcal{N}(\mathbf{0}, \sigma^2), \tag{16}$$

$$\sqrt{n}\left(\widehat{w}_n(a_i, \boldsymbol{x}_i) - w^\star(a_i, \boldsymbol{x}_i)\right) \overset{d}{\to} \mathcal{N}(\mathbf{0}, \sigma^2), \tag{17}$$

where Eq. (16) is due to Slutsky's theorem and Eq. (17) is obtained by substitution of the definition for $w^\star(a_i, \boldsymbol{x}_i)$. The variance is obtained by defining the Softmax function $s(\boldsymbol{\lambda}) = \frac{\exp\left(\boldsymbol{g}_i^\top \boldsymbol{\lambda} + \ell_i\right)}{\frac{1}{n}\sum_{i=1}^n \exp\left(\boldsymbol{g}_i^\top \boldsymbol{\lambda} + \ell_i\right)}$. We denote the gradient of the Softmax function by $\nabla s(\boldsymbol{\lambda})$. We can write (Van der Vaart, 2000, Chapter 3):

$$\sigma^2(a_i, \boldsymbol{x}_i) = \nabla s(\boldsymbol{\lambda}^\star)^\top \, \boldsymbol{V} \, \nabla s(\boldsymbol{\lambda}^\star).$$

Substituting the value of $\boldsymbol{V}$ from (14), we conclude:

$$\boxed{\sigma^2(a_i, \boldsymbol{x}_i) = \nabla s(\boldsymbol{\lambda}^\star)^\top \, \mathbb{E}[\dot{\boldsymbol{\psi}}_{\boldsymbol{\lambda}^\star}]^{-1}\mathbb{E}[\boldsymbol{\psi}_{\boldsymbol{\lambda}^\star}\boldsymbol{\psi}_{\boldsymbol{\lambda}^\star}^\top]\mathbb{E}[\dot{\boldsymbol{\psi}}_{\boldsymbol{\lambda}^\star}]^{-1} \, \nabla s(\boldsymbol{\lambda}^\star).}$$

Note that the value of the softmax function depends on the value of $(a_i, \boldsymbol{x}_i)$ at each point. □

B. Neural Network and Training Details

B.1. Details of the ℓ_θ Neural Network

The $\ell_{\boldsymbol{\theta}}$ network is defined as follows:

$$\ell_{\boldsymbol{\theta}}(z) = cz + \text{dense3}(\,\text{elu}(\,\text{layer\_norm}(\,\text{dense2}(\,\text{tanh}(\,\text{dense1}(\,z\,)\,)\,)\,)\,)\,)$$

The linear term cz acts as a skip connection. The input and output dimensions for the dense linear layers are as follows:

$$\begin{aligned}\text{dense1} :& 1 \mapsto h,\\ \text{dense2} :& h \mapsto h,\\ \text{dense3} :& h \mapsto 1,\end{aligned}$$

where h denotes the hidden dimension. Because the softmax function is invariant to the constant shifts, we do not have any bias terms for dense3 and the skip connection. dense2 also does not have the bias because of the proceeding layer normalization. The dimension h has been tuned as a hyperparameter on a validation data and set to 10.

B.2. Details of the Propensity Score Computation for IPW

We model both $f(a)$ and $f(a|\boldsymbol{x})$ as univariate normal distributions. This is the correct assumption in our synthetic data. The marginal distribution $f(a)$ is estimated by simply finding the mean and standard deviation of the observed treatment values. For the conditional distribution, we write $\mathrm{a}|\mathbf{x} \sim \mathcal{N}(\mu_a(\mathbf{x}), \sigma^2_{a|\boldsymbol{x}})$, where $\mu_a(\mathbf{x})$ is modeled using a feedforward neural network with two layers and $\sigma^2_{a|\boldsymbol{x}}$ is estimated using the residuals of the neural network predictions. The dimension of the neural network has been tuned as a hyperparameter on validation data and set to 30.

B.3. Further Training Details

We used PyTorch to implement E2B. For reproducibility purposes, we provide the final settings used for training:

- Learning algorithm: Adam with learning rate 0.001, no AMSGrad.
- Batch size: 100
- Max epochs: 400
- Weight decay: 2.5×10^{-5}.
- Validation on a dataset of size 400, every 10 steps.

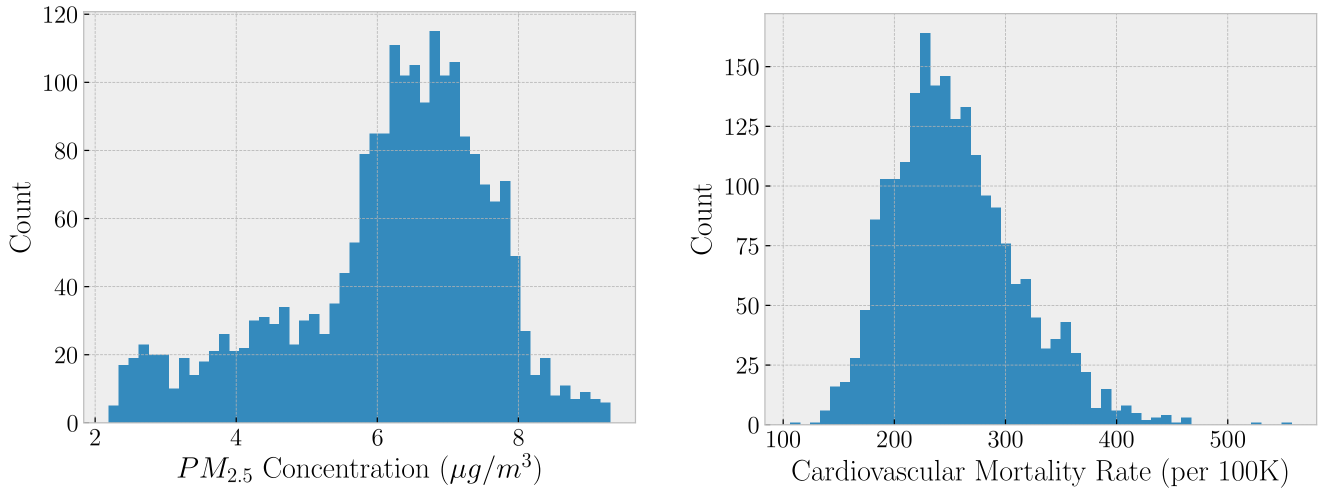
B.4. Detail of Permutation Weighting

We created a stacked data by stacking $\{(\boldsymbol{x}_i, a_i, a_i \odot \boldsymbol{x}_i)\}_{i=1}^n$ and $\{(\boldsymbol{x}_i, \widetilde{a}_i, \widetilde{a}_i \odot \boldsymbol{x}_i)\}_{i=1}^n$, where $\widetilde{a}$ are permutations of the original treatments. We trained a random forest classifier to predict whether each data is from the permuted or the original set. We tried both random forests and neural networks and obtained better results with the former. We also calibrated the predicted probabilities of the classifier before computation of the weights.

C. Data and Preprocessing Description

Table 2: NSAPH Data Description

	PM2.5	CMR	healthfac	population	ses	unemploy	HH_inc	femaleHH	vacant	owner_occ	eduattain	pctfam_pover
count	2132.0	2132.0	2132.0	2132.0	2132.0	2132.0	2132.0	2132.0	2132.0	2132.0	2132.0	2132.0
mean	6.17	255.25	0.18	10.78	0.0	7.85	10.69	11.92	14.25	71.44	35.03	11.25
std	1.45	56.76	0.5	1.26	0.96	2.83	0.24	3.94	8.71	7.76	7.07	5.2
min	2.19	106.14	-2.85	6.2	-1.84	0.0	9.91	2.1	3.8	19.3	9.4	0.0
25%	5.51	215.38	0.0	10.04	-0.67	6.0	10.54	9.3	8.8	67.7	30.4	7.6
50%	6.43	248.16	0.14	10.62	-0.14	7.6	10.67	11.2	11.65	72.7	35.4	10.55
75%	7.15	288.77	0.34	11.46	0.47	9.3	10.83	13.6	16.6	76.7	39.9	13.82
max	9.3	557.43	3.33	16.07	6.46	30.9	11.66	38.0	74.0	89.7	54.6	44.9



(a) Histogram of $PM_{2.5}$

(b) Histogram of Cardiovascular Mortality Rate

Figure 3: The Histograms of Data