



DOUBLY ROBUST OPTIMAL TREATMENT REGIME VIA NEURAL NETWORKS

RAUL T. ARAGON¹, AO YUAN^{1,*}, AND MING T. TAN^{1,*}¹*Department of Biostatistics, Bioinformatics & Biomathematics, Georgetown University, Washington DC, 20057, USA*

ABSTRACT. The optimal treatment regime (OTR) is a recent development in personalized medicine. It assigns the best treatment to an incoming patient according to his/her covariate profile. Construction of the optimal regime hinges on unbiased estimates of the treatment effects. In observational studies, it is known that traditional estimates of treatment effects are biased, and causal inference methods are needed to construct unbiased estimates. The doubly robust estimation (DRE) is a recent popular causal inference procedure for this purpose. It requires the specification of a propensity score model and an outcome model, yielding unbiased estimates as long as one of the models is correctly specified. However, when both models deviate from the true ones even slightly, the DRE is still biased, which can be the case if both models are specified parametrically. Here, we consider observational studies with large-enough data for the more robust nonparametric neural network model to work well but not large enough for deep learning. We develop OTR using DRE from neural networks. Extensive simulation studies are carried out to evaluate the performance of the proposed method, and compare it against its parametric counterpart. Our findings show that the resulting OTR outperforms that under the parametric model when it is incorrectly specified.

Keywords. Causal effect, doubly robust estimation, neural network, optimal treatment regime.

© Journal of Decision Making and Healthcare

1. INTRODUCTION

Optimal treatment regime (OTR) design has received extensive attention in clinical trials due to its practical significance and promise to optimize the therapy, e.g., in mental health, ADHD, and cancer. It is a rule to assign the best treatment to a subject, among several treatments, according to the subject's baseline profile, so as to maximize the subject's benefit from the treatment. Extensive research has been proposed to achieve this goal. The early works include [25, 36, 21, 3] proposed A -learning; [39] studied the inverse propensity score weighted (IPSW) method; [9] considered concordance-assisted learning; More recently, [5] proposed a semiparametric method targeting the treatment difference value function; and [23] considered penalized basis methods.

In OTR construction, unbiased estimates of treatment effects are needed. In observational clinical studies, where treatment assignment is not completely random nor balanced, traditional estimates of treatment effect will be biased due to confounding and/or selection bias. Thus, causal inference methods are needed to obtain unbiased estimates. The inverse propensity score weighting method (IPSW) [14] is one of the earliest such methods to adjust the bias. [28, 27] laid the theoretical foundation for this field (now known as causal inference). Further works include [19], etc. However, if the propensity score model is incorrectly specified, the IPSW estimator can still be biased. The doubly robust estimator (DRE) is a recent further development in causal inference. It requires the specifications of a propensity score model and an outcome model, and the estimator is unbiased as long as either model is correctly specified ([26, 30]). DRE is no panacea, however, since there are problems when the propensity

*Corresponding author.

E-mail address: rdt49@georgetown.edu (R. T. Aragon), ay312@georgetown.edu (A. Yuan), and mtt34@georgetown.edu (M. T. Tan)

Received: July 21, 2026; Revised: July 22, 2026; Accepted: August 5, 2026.

2020 Mathematics Subject Classification: 62P10, 62F35.

score is close to 0 or 1. Observing this issue, [6, 18], proposed possible solutions. Similarly, [15, 31] provide a comprehensive review of DRE, and point out potential drawbacks if both propensity model and outcome model are misspecified. The method in [9] is double robust but is based on rank which does not maximize the value function, and it is not efficient. This method is not robust enough as many of the methods that use parametric models. Other more robust methods, such as [5, 37, 38, 40] used semiparametric models rendering them very robust. These methods, however, focus mainly on the estimation of treatment effects, not on the optimal treatment regime.

Also, most existing methods use parametric models, which are not robust. The semiparametric models are more robust. In contrast, the nonparametric methods are most robust, which include the machine learning (ML) methods such as neural network models, support vector machine [34], or ensemble methods such as random forest [4] can be used to predict the label or class in discrete-dependent variable cases. One of the challenges with these nonparametric algorithms is their interpretability. The resulting parameters obtained after training do not really convey a sense to characterize the contributions of each input variable. To address this issue, [4] proposed Mean Decrease Impurity and, later, Mean Decrease Accuracy as metrics for assessing variable importance. [35] proposed recursive elimination to assess variable importance, but none of these methods provides a measure of importance, just a ranking. Hence, it is difficult to use these models to establish optimal treatment regimes based on the baseline profile.

Recent advances in computation have resulted in the proliferation of neural network models. First introduced in the 1940s [20], neural networks mimic the brain's communication networks to process information and make predictions. Consider a network consisting of nodes connected by edges. Information flows from the first layer of nodes (input layer) to the second layer of nodes through the edges. At each node, information coming in from another node becomes the input of an activation function that determines whether the information should stall or proceed to the next node. If the predicted output does not match the ground truth set by the training label, the information flows back, and the weights that determine how much a node should fire are then updated. (The algorithm that performs this updating is the backpropagation algorithm [29]). The information flows forward again and the process is repeated until the discrepancies between the labeled data and the predicted label fall below a previously specified threshold called the learning rate.

For observational studies or any nonrandomized trials, we propose a non-parametric, doubly robust optimal treatment regime estimator that maximizes the doubly robust estimated value function for moderately sized datasets (hundreds of observational units), which is practical in today's clinical studies. This estimator relies on neural networks for estimating both, (1) the propensity to receive treatment, and (2) the outcome models to estimate the treatment effects. The neural network model works well for moderately sized datasets, but may not perform well with sample sizes in the low hundreds. Our simulation study (described below) suggests that for sample sizes between 400 to 1000, and for a small-dimensional design matrix ($X \in \mathbb{R}^p$ with $2 < p < 13$), one hidden layer performed just as well (or better) than two hidden layers. If the sample size is very large, the nonparametric models can estimate the true parameters very well, then we can just use the outcome models to estimate the mean treatment effects well with no need for DRE. Here we consider the more practical case of a moderately large sample size, where nonparametric estimation is insufficient —warranting the use of DRE. We find that other hyperparameters, such as the number of nodes in the hidden layer or the learning rate, have a greater impact on unbiasedness. Compared with the existing parametric DRE, the proposed method shows an apparent advantage, especially when the functional form of the outcome model is misspecified, or the number of levels in the treatment variable is more than two.

2. THE METHOD

We describe the proposed method for a single-stage observational/nonrandomized clinical trial. Consider the observed data $D_n = \{(y_i, \mathbf{x}_i, a_i) : i = 1, \dots, n\}$ i.i.d. $(y, \mathbf{x}, a)$, where for the i -th individual, y_i is the response, $a_i \in \{1, \dots, k\}$ is the indicator of assigned treatment, $\mathbf{x}_i$ is a vector of covariates that includes baseline values as well as a treatment effect modifier. The goal is to use the observed data to estimate the model parameters and design an optimal treatment assignment rule for a future patient based on her covariate profile.

The response can also be written as $y_i = I_{\{a_i=1\}}y_i^{(a_i=1)} + \dots + I_{\{a_i=k\}}y_i^{(a_i=k)}$, where $y_i^{(a_i=j)}$ is the potential response with treatment j (only one of them is observed, and the rest are potential and latent).

As stated above, a treatment regime is a set of decision rules $d(\mathbf{x})$ i.e., a treatment assignment for a subject with covariate $\mathbf{x} \in \mathcal{X}$. A subject receives treatment j if $d(\mathbf{x}) = j$ and thus her potential outcomes can be expressed as $d(\mathbf{x})y^{(a=d)} + d(\mathbf{x})y^{(a=1)} + \dots + d(\mathbf{x})y^{(a=k)} = I_{(a=1)}Y^{(a=1)} + \dots + I_{(a=k)}Y^{(a=k)}$. Let $G\{y^{(a=d)}\}$ be a given criterion function of interest, a function of realized potential outcome $y^{(a=d)}$. Different criterion functions can be used according to practical problems.

For example, in the simplest case $G(\cdot)$ is the identity function, so $G\{y^{(a=d)}\} = y^{(a=d)}$. Or perhaps, we assume y represents the number of days without a schizophrenic incident or drug abuse relapse; then interest might be in the average number of days without an incident under certain treatment d : $G\{y^{(a=d)}\} = N_j^{-1} \sum_{j=1}^{N_j} y_{ji}^{(a=d)}$. Let $V(d) = \mathbb{E}(G\{y^{(a=d)}\})$ be the value of treatment assignment d , which can be well estimated from the observed data. Let $\mathcal{D} = \{1, \dots, k\}$ be the set of all possible treatment decision rules d . An optimal treatment regime is a treatment assignment rule $d^* \in \mathcal{D}$ for unobserved incoming patients (not for the patients with observed outcomes and for whom the treatment is already assigned), such that $V(d^*) \geq V(d)$ for all $d \in \mathcal{D}$. It then follows that

$$d^*(\mathbf{x}) = \arg \max_j \{\mathbb{E}[G(y)|a = j, \mathbf{x}] : j = 1, \dots, k\}.$$

The above is also observed in [1] for binary treatment assignment.

Denote $g_a(\mathbf{x}) = \mathbb{E}\{G(y)|a, \mathbf{x}\}$, then $d^*(\mathbf{x}) = \arg \max_j \{g_j(\mathbf{x}) : j = 1, \dots, k\}$. Given $g_a(\cdot)$, for a decision point rule d , in case of completely randomized clinical trial without censoring, its value $V(d)$ would be estimated by its empirical version

$$V_n(d) = \frac{1}{n} \sum_{i=1}^n (I_{[d(\mathbf{x}_i)=1]}g_1(\mathbf{x}_i) + \dots + I_{[d(\mathbf{x}_i)=k]}g_k(\mathbf{x}_i))$$

and the value function at d^* is estimated by $V_n(d^*)$.

However, as pointed out by [39, 1], the above method can lead to an estimate far from the optimal regime when the model is misspecified. Furthermore, $g_a(\cdot)$ is unknown and must be estimated along with the propensity score, especially in the case of a practical situation of a non-randomized clinical trial. For this reason, we use the doubly robust estimate of $V(\cdot)$ described below.

2.1. Doubly robust estimation. Based on the work of Yuan et al. (2021), we propose fitting neural networks to estimate both propensity and outcome models within the doubly robust estimation framework to then obtain the doubly robust estimate of the value function $V(\cdot)$.

We first give a brief review of the DRE. Causal inference methods help us recover the true average treatment effect (ATE) in the absence of randomization. Consider a dataset $D_n = (y_i, \mathbf{x}_i, a_i)$ where a_i was not assigned at random, but based on observed covariates $\mathbf{x}_i$. The inverse probability score

We extend it to the dynamic treatment (multi-stage trial) case in a follow-up paper.

There are no strict guidelines as to how long $\mathbf{x}_i$ should be. A neural network with one hidden layer can still learn non-linear patterns that ordinary least squares would not [7].

weighted estimate (IPSW) adapts inverse probability weighting from survey sampling [14]) to causal inference via propensity scores [27], allowing unbiased estimation of ATE as if treatment had been randomized. The validity of the method hinges on correctly specifying a model for the treatment assignment $\pi(\mathbf{x}_i)$, the propensity score. For $A_i \in \{0, 1\}$, typically, $\pi(\mathbf{x}_i)$ is estimated with a logistic model:

$$\pi(\mathbf{x}_i|\beta_A) := P(A_i = 1|X_i = \mathbf{x}_i) = \frac{\exp(\mathbf{x}_i^\top \beta_A)}{1 + \exp(\mathbf{x}_i^\top \beta_A)}$$

Armed with the propensity scores for each observation, we can then estimate $\text{ATE} = \mathbb{E}(Y^{(a=1)} - Y^{(a=0)}) = \mu_1 - \mu_0$ where $\hat{\mu}_1 = n^{-1} \sum_{i=1}^n a_i y_i / \pi(\mathbf{x}_i)$ is an unbiased estimate of μ_1 and $\hat{\mu}_0 = n^{-1} \sum_{i=1}^n (1 - a_i) y_i / (1 - \pi(\mathbf{x}_i))$ is an unbiased estimate of μ_0 [27].

As with IPSW, DRE relies on a propensity score model. However, it allows for the model to be misspecified as long as a secondary model (outcome) is correctly specified [24]. In other words, the method is robust to either misspecification of the propensity score model or of the outcome model. Hence it is named “doubly robust.” Only one of the two must be correctly specified.

One has to define an outcome model per level of treatment. For the $A_i \in \{0, 1\}$ case, we have $y_i|(a_i = 1, \mathbf{x}_i) = m_1(\mathbf{x}_i^\top \beta_{1Y}) + \varepsilon_i$ where $\mathbb{E}(\varepsilon_i|\mathbf{x}_i) = 0$, and similarly for $y_i|a_i = 0, \mathbf{x}_i$. Maximum Likelihood (MLE) is typically used for estimating β_Y . Then, under conditional exchangeability, $\mathbb{E}(Y_i^{(a=1)} - Y_i^{(a=0)}) = \mathbb{E}(Y_i|A = 1) - \mathbb{E}(Y_i|A_i = 0) := \mu_1 - \mu_0$ where

$$\hat{\mu}_1 = n^{-1} \sum_{i=1}^n \left(g_1(\mathbf{x}_i^\top \beta_{1Y}) + \frac{a_i(y_i - g_1(\mathbf{x}_i^\top \beta_{1Y}))}{\pi_1(\mathbf{x}_i^\top \beta_{1A})} \right) \quad (2.1)$$

$$\hat{\mu}_0 = n^{-1} \sum_{i=1}^n \left(g_0(\mathbf{x}_i^\top \beta_{0Y}) + \frac{(1 - a_i)(y_i - g_0(\mathbf{x}_i^\top \beta_{0Y}))}{1 - \pi_1(\mathbf{x}_i^\top \beta_{0A})} \right) \quad (2.2)$$

are the DRE unbiased estimators of μ_1 and μ_0 respectively with $g_a(\cdot)$ as the outcome model and $\pi_1(\cdot)$ as the propensity model.

Equations (2.1) and (2.2) can be generalized to handle multi-level treatments as follows:

$$\hat{\mu}_j = n^{-1} \sum_{i=1}^n \left(g_j(\mathbf{x}_i^\top \beta_{jY}) + \frac{I_{a_i=j}(y_i - g_j(\mathbf{x}_i^\top \beta_{jY}))}{\pi_j(\mathbf{x}_i^\top \beta_{jA})} \right) \quad (2.3)$$

Note that g_j does not contain A_i explicitly. Information about A_i is encoded in $\hat{\mu}_j$ (not g_j) in the propensity scores as well as in what observations go in the numerator, but not in g_j . Estimation of β_{jY} depends on the functional form of g_j , the outcome model. Typical forms include exponential ($y_i = \exp(\mathbf{x}_i^\top \beta_Y) + e_i$) or linear ($y_i = \mathbf{x}_i^\top \beta_Y + e_i$) where $\mathbb{E}(e_i) = 0$.

We propose using a neural network model for the g_j 's. As neural networks are nonparametric models, they are flexible and can approximate more functional forms better than the parametric model or linear models, and with relatively large sample size they provide better model estimation. We now turn to the neural network model.

2.2. Neural networks. Following the notational convention introduced by [29] and adopted in [11], let $\mathbf{x}$ be an input vector. Then, for g_a of a fixed a , a neural network model of L hidden layers with activation function f can be recursively expressed as

$$\begin{aligned} g_a(\mathbf{x}|\theta_a^{(o)}) &= \mathbb{E}[y_i|\mathbf{x}_i, a_i = a] \\ &= f_{a,L}(W_{a,L} \cdot f_{a,L-1}(W_{a,L-1} \cdot \dots \cdot f_{a,1}(W_{a,1} \cdot \mathbf{x} + \mathbf{b}_{a,1}) + \dots \mathbf{b}_{a,L-1}) + \mathbf{b}_{a,L}), \end{aligned} \quad (2.4)$$

In the absence of randomization, $\mathbb{E}(Y^{(a)}) \neq \mathbb{E}(Y|A = a)$ due to confounding. Conditional exchangeability states that when all confounding paths are blocked or controlled for by X , then potential outcome $Y^{(a)}$ is equivalent in expectation to $(Y|A = a)$

where $W_{a,L}$ is a matrix of weights connecting the nodes of layer L with the output coming out of layer $L - 1$; $\mathbf{b}_{a,L}$ is the intercept vector shifting $f_{a,L}(\cdot)$; and $f_{a,L}(\cdot)$ is the activation function.

The dimension of W_l depends on the number of nodes in the previous layer and the number of nodes in the layer it represents. The number of nodes or units for the input layer is equal to the dimension of $\mathbf{x}$ by design. The subsequent layers can have two or more nodes. A neural network model with three or more layers is called a deep learning model [11, 17]. There is a vast literature relating to activation functions [22, 8]. Common choices are the logistic and rectified linear unit (ReLU). ReLU tends to be favored because it avoids gradient explosion (output values that are very large real numbers) or gradient leakage (output values that are numerical zero) [10].

Various architectures have been proposed to handle different tasks. For example, if the data are highly nonlinear or hierarchical, such as in image classification or natural language processing, deep networks generally perform better than single hidden layer networks [2]. The proliferation of “big data” has required a similar proliferation of deep network architectures that not always involve fully connected layers, but instead, involve convolutions as opposed to matrix multiplication. (See [16, 13, 32])

The doubly robust estimation functions we approximate with neural networks in this work do not require complex architectures nor convolutions. Selection of number of nodes per hidden layer(s) and activation function can be accomplished via k-fold cross-validation [33, 12]. (See simulation section for more details on the tuning of the neural networks we use in this study.)

2.2.1. The propensity score model with neural networks. As stated above, DRE requires at least two model families: a propensity score model and an outcome model. The propensity score model is typically a parametric logistic model (or multinomial in the multi-level treatment case), where $\mathbb{P}(a = j|\mathbf{x}) = \exp(\mathbf{x}^\top \boldsymbol{\beta}) / (1 + \exp(-\mathbf{x}^\top \boldsymbol{\beta}))$. Here, we use neural network model: Using the data $\{(a_i, \mathbf{x}_i) : i = 1, \dots, n\}$, we assume the propensity score model

$$P(a = j|\mathbf{x}) = \pi_j(\mathbf{x}|\theta) \\ := f_{a,L}^{(p)}(W_{a,L}^{(p)} \cdot f_{a,L-1}^{(p)}(W_{a,L-1}^{(p)} \cdot \dots \cdot f_{a,1}^{(p)}(W_{a,1}^{(p)} \cdot \mathbf{x} + \mathbf{b}_{a,1}^{(p)}) + \dots \mathbf{b}_{a,L-1}^{(p)}) + \mathbf{b}_{a,L}^{(p)}), \quad (2.5)$$

where (p) indicates the matrix (or function) corresponds to the ‘propensity’ model, $W_{a,1}^{(p)} \in \mathbb{R}^{|\mathbf{x}| \times |L_1|}$, and the cardinality of L_1 is to be determined by the data in cross-validation. The number of nested functions (i.e. layers) and the form of $f(\cdot)$ (e.g. ReLU or sigmoid) are also hyperparameters to be chosen via cross-validation.

2.2.2. The outcome model with neural networks. Similarly, we use neural networks to represent the outcome model g_j in equation (2.3). Hence the mean treatment effect under treatment j is consistently estimated by

$$\hat{\mu}_j = n^{-1} \sum_{i=1}^n \left(g_j(\mathbf{x}_i) + \frac{I_{a_i=j}(y_i - g_j(\mathbf{x}_i))}{\pi_j(\mathbf{x}_i)} \right) \quad (2.6)$$

where $g_j(\mathbf{x}_i) := f_{a,L}^{(o)}(W_{a,L}^{(o)} \cdot f_{a,L-1}^{(o)}(W_{a,L-1}^{(o)} \cdot \dots \cdot f_{a,1}^{(o)}(W_{a,1}^{(o)} \cdot \mathbf{x} + \mathbf{b}_{a,1}^{(o)}) + \dots \mathbf{b}_{a,L-1}^{(o)}) + \mathbf{b}_{a,L}^{(o)})$ and the row dimension of $W_{a,L}^{(o)}$, the number of nested activation function $f_{a,L}^{(o)}(\cdot)$, and the form of $f_{a,L}^{(o)}(\cdot)$ are all hyperparameters to be chosen via cross-validation from the data.

2.3. OTR by DRE via ML. The method we propose aims to find an OTR ($\mathbf{d}^*$) that a new patient must follow at each decision point k as to maximize (or minimize) some outcome metric Y or a function $H(y)$. For simplicity, we assume $H(\cdot)$ is the identity function. Then, at decision point $k = 1$,

$$V(d) = \mathbb{E}[H(y)] = \mathbb{E}\{\mathbb{E}[Y|A = d, X = \mathbf{x}]\}$$

and

$$d^*(\mathbf{x}) = \arg \max_a \mathbb{E}\{\mathbb{E}[Y|A = a, X = \mathbf{x}]\}.$$

If randomization exists at decision point $k = 1$, or if the assignment mechanism is known, $d^*(\mathbf{x})$ can be easily found by optimizing the empirical counterpart $V_n(d)$ of $V(d)$

$$V_n(d) = n^{-1} \sum_{i=1}^n \sum_{j=1}^k I_{d(\mathbf{x}_i)=j} \mathbb{E}(Y_i^{a=j}),$$

where $\mathbb{E}(Y^{a=j})$ under randomization is equivalent to $\mathbb{E}(Y_i|A_i = j) := \mu_j$. However, as pointed out, in the absence of randomization, but under conditional exchangeability, we need causal inference methods to estimate $\mathbb{E}(Y|A = a_i, X = \mathbf{x}_i)$. To that end, let $\hat{\mu}_j$ be the DRE estimate of $\mathbb{E}(Y|A = a_i, X = \mathbf{x}_i)$.

2.3.1. DRE of $V(d)$ and d^* in a single-stage treatment regime. Let $z_i = G(y_i)$ (which simplifies to y_i when $G(\cdot)$ is the identity function), $\delta_{ij} = I_{(a_i=j)}$, and $\mu_j = E[G(y_i)|a_i = j, X = \mathbf{x}_i]$ be the mean of $G(Y^{(a=j)})$ under treatment j . To make the discussion concrete, assume $\mu_j > 0$ for $j = 1, \dots, k$ and that a higher $G(Y)$ is desirable. After obtaining the propensity score estimates $\hat{\pi}_{a_i}(\cdot)$ and the outcome model estimates $\hat{g}_{a_i}(\cdot)$ for each $i \in \{1, \dots, n\}$, the DRE of μ_j is (Yin, Yuan and Tan, 2024)

$$\hat{\mu}_j = \frac{1}{n} \sum_{i=1}^n \left(\hat{g}_j(\mathbf{x}_i) + \frac{\delta_{ij}(z_i - \hat{g}_j(\mathbf{x}_i))/\hat{\pi}_j(\mathbf{x}_i)}{n^{-1} \sum_{i=1}^n \delta_{ij}/\hat{\pi}_j(\mathbf{x}_i)} \right) := \frac{1}{n} \sum_{i=1}^n \hat{\mu}_j(\delta_{ij}, z_i, \mathbf{x}_i) \quad (2.7)$$

for $(j = 1, \dots, J)$ and for fixed $d(\cdot)$, the DRE of $V(d)$ is

$$\begin{aligned} V_n(d) &= \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k I_{(d(\mathbf{x}_i)=j)} \hat{\mu}_{n,j}(\delta_{ij}, z_i, \mathbf{x}_i) \xrightarrow{a.s.} \mathbb{E} \left[\sum_{j=1}^k I_{(d(\mathbf{X})=j)} \mu_j(A, Z, \mathbf{X}) \right] \\ &= \mathbb{E} \left[\sum_{j=1}^k I_{(d(\mathbf{X})=j)} \mathbb{E}[\mu_j(A, Z, \mathbf{X}) | \mathbf{X}] \right] \\ &= \mathbb{E} \left[\sum_{j=1}^k I_{(d(\mathbf{X})=j)} \mu_j(\mathbf{X}) \right], \end{aligned} \quad (2.8)$$

where

$$\mu_j(\mathbf{x}) = \mathbb{E}[\mu_j(A, Z, \mathbf{X}) | \mathbf{X} = \mathbf{x}], \quad \mu_j(a, z, \mathbf{x}) = \lim_{n \rightarrow \infty} \hat{\mu}_j(\delta_j, z, \mathbf{x}).$$

Asymptotically, when $d(\mathbf{x}) \in \{0, 1\}$, the optimal regime $d^*(\cdot)$, when a higher $G(y)$ is desirable, in (2.8) is $d^*(\mathbf{x}) = \arg \max_{j \in \mathcal{D}} \mu_j(\mathbf{x})$ (in the sense $d^*(\mathbf{x}) = j$ means $d^*(\mathbf{x}) = I_{a=j}$). The $\mu_j(\cdot)$'s are unknown and are well estimated by

$$\hat{\mu}_{n,j}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \hat{\mu}_j(\delta_{ij}, a_i, \mathbf{x}_i), \quad (j = 1, \dots, k)$$

yielding

$$V_n(d) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^k I_{(d(\mathbf{x}_i)=j)} \hat{\mu}_{n,j}(\mathbf{x}_i). \quad (2.9)$$

Maximizing $V_n(d)$ over d in (2.9), we get the estimated optimal regime as, for any patient with covariate vector $\mathbf{x}$,

$$d_n^*(\mathbf{x}) = j(\mathbf{x}), \quad j(\mathbf{x}) = \arg \max_{j \in \{1, \dots, k\}} \hat{\mu}_{n,j}(\mathbf{x})$$

Or we may use randomized assignment with probability $\hat{p}_j(\mathbf{x}) = \hat{\mu}_{n,j}(\mathbf{x}) / \sum_{r=1}^k \hat{\mu}_{n,j}(\mathbf{x})$ assign to treatment j .

3. SIMULATION STUDY

In this section, we study the performance of DRE via ML (DRE-ML) for finding OTRs with one single stage. We compare its performance against the naive approach (described below) as well as against DRE-parametric when the model matches the true functional form of $P(A_i|\mathbf{x}_i) = \pi(\cdot)$ and Y_i , and when it does not. We assume treatment is discrete, well-defined (SUTVA), and with common support everywhere in the covariate space (positivity). We also assume we observe, and control for, all confounders of the treatment indicator A_i and the outcome Y_i (exchangeability). Without loss of generality, we simulate cases where a higher Y_i is preferable.

We consider the single-stage treatment regime case, where finding $d^*(\mathbf{x}_i)$ amounts to finding the ATE that leads to a maximum Y_i . We compare DRE-ML performance against that of two parametric models: logistic for propensity scores and exponential (DRE-param expo) and ordinary least squares (DRE-param ols) for outcome. Treatment assignment is not random but based on baseline covariates $\mathbf{x}_i$.

All parameters $(\beta_A, \beta_Y, \delta_1, \delta_2)$ for each of the 6 DGPs across all 180 data sets can be access and readily replicated with seed values as presented in the Supplementary Materials section.

3.1. Binary treatment. Let A_i be a binary treatment variable with values $\{0, 1\}$. $P(A = 1|X)$ is governed by the propensity function $\pi(\cdot|X)$, which can be logistic or sigmoid. Let Y_i be the outcome of interest. $Y_i = g(\mathbf{x}_i^\top \beta_Y + \delta_1 A_i + \delta_2 [A_i \times X_{p,i}]) + \varepsilon_i$ where $g(\cdot)$ is either exponential or log-gamma. δ_1 is the main effect of $A_i = 1$, and δ_2 is the interaction effect of A_i and binary ($\{0, 1\}$) tailoring variable $X_{p,i}$. We set $\delta_1 \in \mathbb{R}^+$ and $\delta_2 \in \mathbb{R}^-$ to ensure non-trivial OTR. We set the sample size $n = 200 \times L$ where L is number of unique elements of A_i . We aim for treatment-and-control split close to 50/50.

We simulate a design matrix $X \in \mathbb{R}^{n \times P}$ with $P = 3$.

$$X \sim \text{MVN}(\mathbf{m}, \Sigma)$$

where mean vector $\mathbf{m}$ is chosen at random from $\text{Unif}(-1, 1)$ and covariance Σ is a $P \times P$ matrix with constant variance $\sigma = 1$ but attenuating covariance $\rho^{|p-p'|}$ for $(x_p, x_{p'})$. ρ is chosen at random from $\text{Unif}(0.4, 0.6)$. This makes the X_p covariates not independent.

The last column of X is binarized as follows:

$$X_{p=P} = \begin{cases} 0 & \text{if } X_{p=P} \leq 0 \\ 1 & \text{if } X_{p=P} > 0 \end{cases}$$

For this binary A_i case, we simulate treatment assignment and outcome variables in two different ways: logit-expo and tanh-loggamma.

3.1.1. logit-expo. We first simulate treatment assignment as logistic and outcome as exponential (logit-expo). We simulate assignment to treatment as follows:

$$P(A_i = 1|\mathbf{x}_i) = \pi_1(A_i|\mathbf{x}_i; \beta_A)$$

where $\pi_1(A_i|\mathbf{x}_i; \beta_A) = \text{expit}(\mathbf{x}, \beta_A) = 1/(1 + \exp\{-\mathbf{x}_i^\top \beta_A\})$, and $\mathbf{x}_i$ is the i^{th} row of design matrix X . β_A is a vector of coefficients in $\mathbb{R}^P$ with predetermined values from $\text{Unif}(-1, 1)$.

If the $\text{sgn}(\delta_1) = \text{sgn}(\delta_2)$ (say positive), then the optimal decision rule is to “give everyone the treatment” (or “refuse the treatment to everyone” if negative).

We fix the β_A coefficients per data set of size n , but we let them differ across data sets. Hence, we have 30 different sets of β_A per DGP.

Next, we simulate the outcome random variable Y_i as

$$Y_i = \exp \left(\mathbf{x}_i^\top \boldsymbol{\beta}_Y + \delta_1 A_i + \delta_2 (A_i \times X_{pi}) \right) + \varepsilon_i$$

where $\boldsymbol{\beta}_Y$ is a vector of coefficients in $\mathbb{R}^P$, and it's chosen at random from $\text{Unif}(-1, 1)$ (but different from $\boldsymbol{\beta}_A$), $\varepsilon_i \sim N(0, 0.01)$, $\delta = 0.6$. The binary covariate X_p modifies the treatment and its modification is captured by the interaction coefficient δ_2 , which we let it be negative from $\text{Unif}(1/\sqrt{p}, 1/\sqrt{p})$ and it's constant within data set, but it varies across data sets. The treatment heterogeneity ensures the OTR will not be trivially $d(\mathbf{x}_i) = 1$ for everyone as explained above.

We collect Y, X , and A into a rectangular data set D with dimensions $(L \times n) \times (P + 2) = (400 \times 5)$.

Since we have $L = |A_i| = 2$, where $a_i \in \{0, 1\}$, we have

$$\begin{aligned} \Delta_{0,1} &= \mathbb{E}(Y^{\{a_i=1\}} - Y^{\{a_i=0\}}) \\ &\equiv \mathbb{E}(Y|\mathbf{x}, a = 1) - \mathbb{E}(Y|\mathbf{x}, a = 0) \\ &= \mu_1 - \mu_0 \\ &\approx \hat{\mu}_1 - \hat{\mu}_0 \end{aligned}$$

Following the DRE procedure, for $a_i = 1$, we compute

$$\hat{\mu}_1 = \frac{1}{n} \sum_{i=1}^n \left(\hat{g}_1(\mathbf{x}_i) + \frac{I_{a_i=1}(y_i - \hat{g}_1(\mathbf{x}_i)) / \hat{\pi}(\mathbf{x}_i)}{n^{-1} \sum_{i=1}^n I_{a_i=1} / \hat{\pi}(\mathbf{x}_i)} \right)$$

and for $a_i = 0$:

$$\hat{\mu}_0 = \frac{1}{n} \sum_{i=1}^n \left(\hat{g}_0(\mathbf{x}_i) + \frac{I_{a_i=0}(y_i - \hat{g}_0(\mathbf{x}_i)) / (1 - \hat{\pi}(\mathbf{x}_i))}{n^{-1} \sum_{i=1}^n I_{a_i=0} / (1 - \hat{\pi}(\mathbf{x}_i))} \right)$$

For DRE-ML, we approximate propensity $\pi(\cdot)$ and outcome $g(\cdot)$ functions with neural networks.

$$\hat{g}_j(\mathbf{x}_i; W, \mathbf{b}) = f_{j,L}(W_{j,L} \cdot f_{j,L-1}(W_{j,L-1} \cdot \dots \cdot f_{j,1}(W_{j,1} \cdot \mathbf{x}_i + \mathbf{b}_{j,1}) + \dots \cdot \mathbf{b}_{j,L-1}) + \mathbf{b}_{j,L})$$

and

$$\hat{\pi}(\mathbf{x}_i; W', \mathbf{b}') = f_{j,L}(W'_{j,L} \cdot f_{j,L-1}(W'_{j,L-1} \cdot \dots \cdot f_{j,1}(W'_{j,1} \cdot \mathbf{x}_i + \mathbf{b}'_{j,1}) + \dots \cdot \mathbf{b}'_{j,L-1}) + \mathbf{b}'_{j,L})$$

We tune the neural network model via cross validation with 6 folds. However, given that there are two types of functions we try to approximate, namely propensity and outcome, we tune each model type with a different hyper parameter space.

For binary propensity functions, we tune the following hyperparameters: number of hidden layers (1 or 2), number of units in each hidden layer (10 to 60), number of epochs (40 to 120), learning rate $\{.001, .005, .01\}$, and activation function (ReLU or sigmoid). Given the large size of this parameter space, we perform randomized cross validation with 15 random hyperparameter values for units and 10 random values for epochs. The best performing parameter combination, in terms of the area under the Receiver Operating Characteristic curve (AUC), across all 6 folds, is used to predict A_i , which we then denote $\hat{\pi}_j$ where $j \in \{0, 1, 2, 3, 4\}$ depending on the dimension of A_i .

Similarly, we tune a neural network to approximate the outcome function that generated Y_i . The tuning hyperparameters are the same to the propensity neural network described above, but the hyperparameter space is different. We explore the number of hidden layers (1 or 2), number of units in each hidden layer (10 to 115), number of epochs (75 to 150), learning rate $\{.001, .005, .01\}$, and activation function (ReLU or sigmoid). As with the tuning of the propensity model, we perform randomized cross validation choosing 15 hyperparameter values at random from the hyperparameter space of units,

As with $\boldsymbol{\beta}_A$ we fix $\boldsymbol{\beta}_Y$ per data set but we let it vary across data sets.

TABLE 3.1. Tuning hyperparameter boundaries per model type in binary treatment cases

Hyperparameter	$\hat{\pi}_j$	$\hat{g}_j$
Activation function	ReLU or tanh	ReLU or tanh
Hidden layers	$\{1,2\}$	$\{1,2\}$
Units per layer	$\text{randint}^1(10, 60)$	$\text{randint}(10, 115)$
Learning rate	$\text{loguniform}^2(1e^{-4}, 9e^{-2})$	$\text{loguniform}(1e^{-2}, 1e^{-1})$
Epochs	$\text{randint}(40, 120)$	$\text{randint}(75, 150)$

¹ random package function to randomly chose an integer from $(a$ to $b)$.

and 10 random hyperparameter values from the epochs hyperparameter space. The best performing parameter combination, in terms of root mean squared error (RMSE) across all 6 folds, is used to predict Y_i , which we denote $\hat{g}_j$. Table 3.1 displays the hyperparameter boundaries side-by-side per model.

For DRE-Param, we approximate propensity and outcome functions with logistic and exponential parametric functions, matching the true functional form in some cases as we will see.

Next, we compute a “naive” estimate which naively estimates $\Delta_{0,1}$ by computing the empirical difference in means between the two treatment groups:

$$\hat{\Delta}_{0,1\text{naive}} = \frac{1}{\sum_{i=1}^n I_{\{a_i=1\}}} \sum_{i=1}^n I_{\{a_i=1\}} Y_i - \frac{1}{\sum_{i=1}^n I_{\{a_i=0\}}} \sum_{i=1}^n I_{\{a_i=0\}} Y_i$$

Lastly, we compute the true ATE (given that we know all true parameters and their functional form) as follows:

Given that $A_i \in \{0, 1\}$ and $X_{pi} \in \{0, 1\}$,

$$\begin{aligned} \Delta_{0,1} &= \mathbb{E}(Y_i^{(a_i=1)} - Y_i^{(a_i=0)}) \\ &= \mathbb{E}(Y|A = 1, X) - \mathbb{E}(Y|A = 0, X) \text{ (by conditional exchangeability)} \\ &= P(X_p = 1)[\mathbb{E}(Y|A = 1, X_p = 1) - \mathbb{E}(Y|A = 0, X_p = 1)] \\ &\quad + P(X_p = 0)[\mathbb{E}(Y|A = 1, X_p = 0) - \mathbb{E}(Y|A = 0, X_p = 0)] \\ &= P(X_p = 1) \left(n^{-1} \sum_{i=1}^n \eta(\mathbf{x}_i^\top \beta_Y + \delta + \delta_2) - n^{-1} \sum_{i=1}^n \eta(\mathbf{x}_i^\top \beta_Y) \right) \\ &\quad + P(X_p = 0) \left(n^{-1} \sum_{i=1}^n \eta(\mathbf{x}_i^\top \beta_Y + \delta) - n^{-1} \sum_{i=1}^n \eta(\mathbf{x}_i^\top \beta_Y) \right) \end{aligned}$$

where $\eta(\cdot) = \exp(\cdot)$, the true functional form of Y_i .

We repeat this process 30 times, where the values of $D = \{A_i, X_i, Y_i; \beta_Y, \beta_A, \delta_1, \delta_2\}$ change but the functional structure remains intact.

Figure 3.1 shows boxplots for the difference between the estimates of the naive, DRE-param (with logit-OLS), DRE-param (with logit-expo), and DRE-ML (neural network) models and the true treatment effect for 30 generated datasets under the logit-expo DGP. We can see on the far-left that the naive estimates have high variance and are not centered at 0 as expected. The logit-expo model is centered at zero and has very small variance. This is expected given that the model mimics the true DGP. On the far right, we see the neural network model. Its performance is comparable to the logit-expo model though it is no match for the DRE-param models.

3.1.2. Tanh-loggamma. We now depart from a case where the DRE-param correctly models the true DGP. We generate X the same as before, keeping the sample size the same, but we generate A_i under

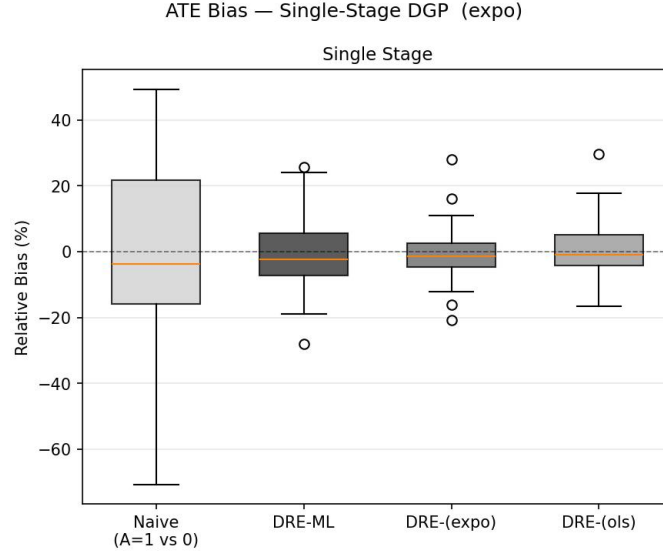


FIGURE 3.1. Difference between $\hat{\Delta}_{0,1}$ and true $\Delta_{0,1}$ across 30 dataset by model under logit-expo DGP

$\tanh(\cdot)$ function and Y_i under log-gamma.

$$A_i \sim \text{Bern}\left(\pi_i = \frac{1}{2}(\tanh(\mathbf{x}_i^\top \boldsymbol{\beta}_A) + 1)\right)$$

where $\boldsymbol{\beta}_A \in \mathbb{R}^P$ chosen from $\text{Unif}(-1, 1)$. We generate Y_i as:

$$Y = C \cdot \frac{\exp\left\{\lambda_1 \cdot \mathbf{x}_i^\top \boldsymbol{\beta}_Y + \delta A_i + \delta_2(A_i \times X_{pi})\right\} \cdot \exp\left\{-\frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta}_Y + \delta A_i + \delta_2(A_i \times X_{pi}))}{\lambda_2}\right\}}{\Gamma(\lambda_1) \cdot \lambda_2^{\lambda_1}} + \varepsilon_i$$

where $\lambda_1 = 3$ and $\lambda_2 = 2$, $\varepsilon_i \sim N(0, 0.1)$, $\mathbf{x}_i$ is the i^{th} row of a design matrix with P columns, $\boldsymbol{\beta}_Y$ is a vector of coefficients in $\mathbb{R}^P$ chosen from $\text{Unif}(-1, 1)$, $C = 10$, and $\delta_1 = \text{Unif}(-1/\sqrt{p}, +1/\sqrt{p})$ and $\delta_2 = -\text{Unif}(-1/\sqrt{p}, +1/\sqrt{p})$.

As before, we generate 30 data sets for each DGP and compute naive, DRE-ML, DRE-param (OLS and expo) estimates of $\Delta_{0,1}$.

Figure 3.2 provides a visual representation of each model's estimates of $\Delta_{0,1}$. It shows boxplots for the difference between the estimates of the naive, DRE-ML, DRE-param (expo), and DRE-param (ols) for 30 generated data sets under the tanh-loggamma DGP. As in the examples before, the naive estimates (left) have high variance and are not centered at 0, as expected. The DRE-ML is the only model providing estimates with zero relative bias.

3.2. Three-level treatment. We now increase the number of treatments to three; that is we allow for $A_i \in \{0, 1, 2\}$ and obtain estimates for $\Delta_{0,1}$ and $\Delta_{0,2}$. We fix the sample size at $n = 600$, roughly 200 per treatment level, though the levels are not always perfectly balanced on each simulation. We simulate a design matrix $X \in \mathbb{R}^{n \times P}$ with $P = 8$ and with one binary column that also acts as the tailoring variable as before.

Because $|A_i| = 3$, several parameters have to be scaled to a higher dimension. We defined a matrix of coefficients $\mathbf{B}_A$ composed of two row vectors of coefficients: $\boldsymbol{\beta}_{A1}$ and $\boldsymbol{\beta}_{A2}$ where the values are taken from $\text{Unif}(1, 1)$ prior to generating each data set.

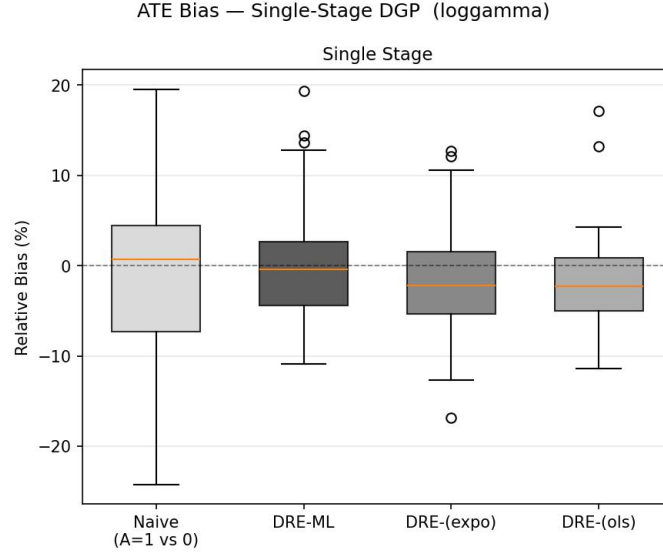


FIGURE 3.2. Difference between $\hat{\Delta}_{0,1}$ and true $\Delta_{0,1}$ across 30 dataset by model under tanh-loggamma DGP

We simulate assignment to treatment based on multinomial logit model:

$$\ln \left(\frac{\pi_j}{\pi_0} \right) = \mathbf{x}_i^\top \boldsymbol{\beta}_{Aj}, \quad (j = 1, 2),$$

for treatment level j .

This allows us to simulate a random vector A_i with probabilities given by

$$\boldsymbol{\pi}(\mathbf{x}|\mathbf{B}_A) = [\pi_j(\mathbf{x}|\boldsymbol{\beta}_{A0}), \pi_{j'}(\mathbf{x}|\boldsymbol{\beta}_{A1}), (1 - (\pi_j(\mathbf{x}|\boldsymbol{\beta}_{A0}) + \pi_{j'}(\mathbf{x}|\boldsymbol{\beta}_{A1})))]$$

where $\mathbf{B}_A$ is a 2×8 matrix of coefficients.

Next, we generate the outcome variable Y_i (after omitting the reference treatment level $a_i = 0$):

$$Y_i = g \left(\mathbf{x}_i^\top \boldsymbol{\beta}_Y + A_i^\top \boldsymbol{\delta}_1 + (A_i \odot X_{pi})^\top \boldsymbol{\delta}_2 \right) + \varepsilon_i$$

where g is either the exponential, log-gamma, log-normal, or sigmoid function defined above. $\boldsymbol{\beta}_Y$ is a vector of coefficients in $\mathbb{R}^P$ and it's chosen at random from $\text{Unif}(-1, 1)$ (but different from $\boldsymbol{\beta}_{Aj}$), $\varepsilon_i \sim N(0, 0.01)$, $\boldsymbol{\delta}_1 = \{|\text{Unif}(-1/\sqrt{p}, 1/\sqrt{p})|, |\text{Unif}(-1/\sqrt{p}, 1/\sqrt{p})|\}$ and the modifier coefficients are $\boldsymbol{\delta}_2 = \{-|\text{Unif}(-1/\sqrt{p}, 1/\sqrt{p})|, -|\text{Unif}(-1/\sqrt{p}, 1/\sqrt{p})|\}$ to ensure non-trivial OTRs. Once the $\boldsymbol{\delta}_r$ values are chosen, they remain constant per data set, but not across data sets. In other words, we select 30 different sets of $\boldsymbol{\delta}_r$. The operator $\odot$ represent element-wise product, which allows us to express interactions for each non-baseline treatment level of A_i , $(A_{1i} \times X_{pi})$ and $(A_{2i} \times X_{p,i})$.

We collect Y, X , and A into a rectangular data set D with dimensions $n \times (P + 3)$.

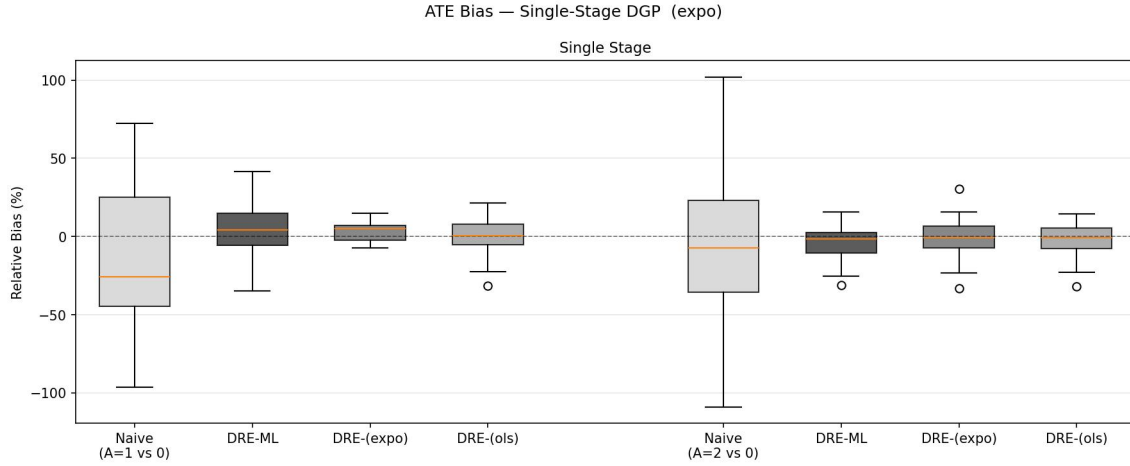
As before, we fit a naive model (empirical mean for each treatment level), DRE-ML with propensity and outcome neural network models as per (2.3), and two DRE-param models: (expo) and (ols). Each model is compared against the true $\Delta_{j,0}$ $j \in \{1, 2\}$.

We repeat this 30 times for 30 different data sets.

Because we have more than two-treatment choices, we tweak the hyper parameter space slightly in tuning.

TABLE 3.2. Tuning hyperparameter boundaries per model type for multi-treatment A_i

Hyperparameter	$\hat{\pi}_j$	$\hat{g}_j$
Activation function	ReLU or tanh	ReLU or tanh
Hidden layers	$\{1, 2\}$	$\{1, 2\}$
Units per layer	<code>randint(15, 60)</code>	<code>randint(5, 115)</code>
Learning rate	<code>loguniform($1e^{-4}$, $1e^{-2}$)</code>	<code>loguniform($5e^{-3}$, $1e^{-1}$)</code>
Epochs	<code>randint(50, 100)</code>	<code>randint(50, 125)</code>

FIGURE 3.3. Difference between $\hat{\Delta}_{j,j'}$ and true $\Delta_{j,j'}$ across 30 datasets by model

3.2.1. *Logit-expo*. We start with 30 data sets under $g(\cdot) = \exp(\cdot)$ and report the results in side-by-side boxplots per treatment contrast from the baseline $A_i = 0$ in figure 3.3. Here we see the parametric model still has an edge on DRE-ML but only when the outcome model matches the true DGP. DRE-Param OLS exhibits higher variance than DRE-Param Expo and (in the A_{01} contrast) DRE-ML.

3.2.2. *Tanh-loggamma*. To remove the edge that DRE-param (expo) has when it matched the true DGP in the previous example, we now simulate 30 data sets under the same conditions as the previous example, except that $\pi(A_i = j|\mathbf{x}; \beta_{A,j}) = \exp(\mathbf{x}^\top \beta_{A,j}) / \sum_{j=1}^2 \exp(\mathbf{x}^\top \beta_{A,j})$ and $g(\cdot) = \log\text{-gamma}(\cdot)$.

To get a better idea of each model's performance, as in previous examples, we present box plots showing the relative bias of $\hat{\Delta}_{j,j'}$ per model in figure 3.4. We can see that the DRE-ML has caught up in terms of smaller variance with DRE-param (expo). The neural-network model seems more consistent. Figure 3.4 also suggests that when the DRE-param does not match the DGP, DRE-ML is a better choice. DRE-param OLS, while also unbiased, also exhibits higher variance than DRE-ML.

3.3. **Five-level treatment.** To conclude the single-stage case, we now present results under 30 data sets where the treatment variable has five levels. Mimicking the previous examples, we first show simulation results when DRE-param matches the DGP (multinomia-expo), followed by results where DRE-param does not match the DGP (tanh-loggamma).

We fixed the sample size at $n = 1,000$, roughly 200 per treatment level. We simulate a design matrix $X \in \mathbb{R}^{n \times P}$ with $P = 12$ and, as before, with one binary column that also serves as the tailoring variable.

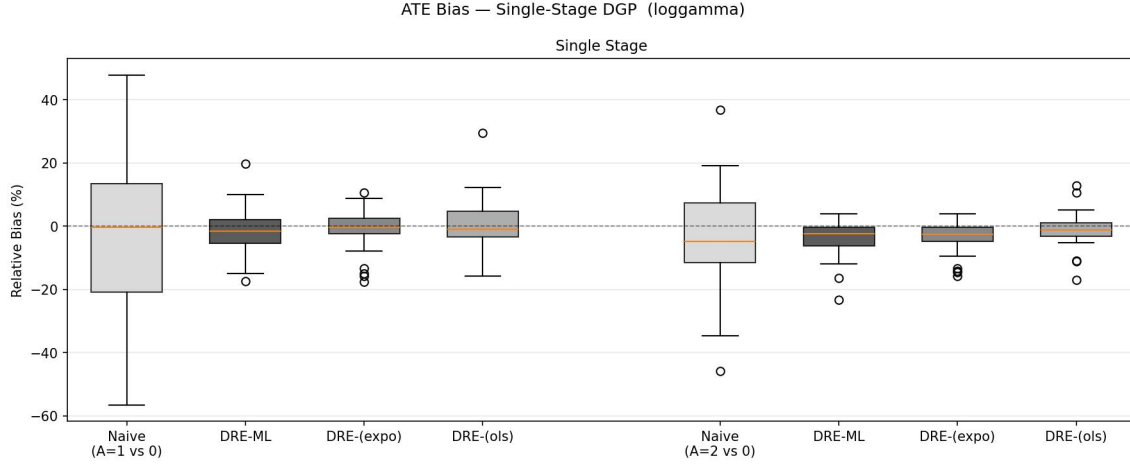


FIGURE 3.4. Difference between $\hat{\Delta}_{j,j'}$ and true $\Delta_{j,j'}$ across 30 datasets by model

As in the three-level treatment case, we create a matrix of parameters $\mathbf{B}_A$ with row vectors β_{A_j} for $j = \{1, 2, 3, 4\}$. Thus, $\mathbf{B}_A \in \mathbb{R}^{4 \times 12}$ where each value is chosen from $\text{Unif}(-1, 1)$ at the beginning of each data set simulation.

The simulation of Y_i mimicks previous examples. That is, we generate

$$Y_i = g\left(\mathbf{x}_i^\top \beta_Y + A_i^\top \delta_1 + (A_i \odot X_{pi})^\top \delta_2\right) + \varepsilon_i$$

where $g(\cdot)$ can be either exponential or log-gamma. We set the values of δ_1 and δ_2 with $\text{Unif}(1/\sqrt{p}, 1/\sqrt{p})$ ensuring the first set is positive while the second set is negative to induce non-trivial OTRs.

3.3.1. Logit-expo. We first consider the case where DRE-param expo matches the DGP. We simulate 30 data sets as described above with $\ln\left(\frac{\pi_j}{\pi_0}\right) = \mathbf{x}_i^\top \beta_{A_j}$ for $j = \{1, 2, 3, 4\}$ and such that $\sum_{j \in J} \pi_j = 1$. We let $g() = \exp()$. We expect the performance of both models to be comparable, with the DRE-param having a slight edge over DRE-ML due to the DRE-param matching the true DGP.

Figure 3.5 shows side-by-side boxplots of the relative-bias across all 30 data sets per model. We can see that DRE-param (expo) achieves the lowest variance across all models overall. DRE-ML comes in second with DRE-param (ols) third. The five-level complexity, though, does not give DRE-param (expo) a clear advantage despite matching the DGP.

We next explore the single-stage case with five-level treatment when the DRE-Param does not match the true DGP.

3.3.2. Tanh-loggamma. To prevent the DRE-param from matching the DGP, we simulate 30 data sets where $\pi(A_i = j | \mathbf{x}; \beta_{A,j}) = \exp(\mathbf{x}^\top \beta_{A,j}) / \sum_{j=1}^4 \exp(\mathbf{x}^\top \beta_{A,j})$ for $j = \{1, 2, 3, 4\}$ subject to $\sum_{j \in J} \pi_j = 1$. We set $g() = \text{log-gamma}()$.

Figure 3.6 shows side-by-side boxplots of the relative-bias per model across all 30 data sets. Here, we can see that DRE-ML is the model with the lowest variance, particularly with $\hat{\Delta}_{0,1}$ and $\hat{\Delta}_{0,4}$. DRE-Param (expo) and DRE-Param (ols) achieve a similar variance with, perhaps, DRE-Param (expo) being a little lower overall.

This single-case simulation study suggests that when the model does not match the DGP, DRE-ML is a better choice for computing precise ATEs with a single-stage DTR with the discrete treatment variable A_i having more than two levels. DRE-Param performed equally well when matching the DGP under binary treatment. However, as more levels of treatment were added, and the functional form

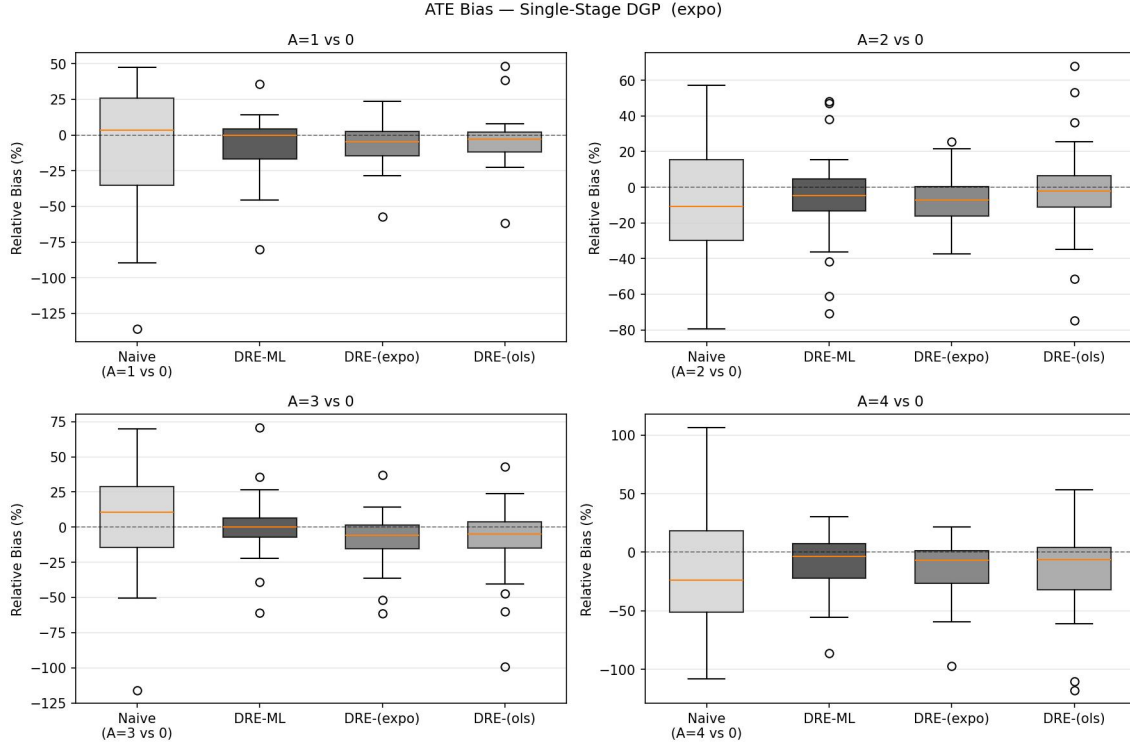


FIGURE 3.5. Difference between $\hat{\Delta}_{j,j'}$ and true $\Delta_{j,j'}$ across 30 datasets by model

was misspecified, DRE-ML showed an advantage. It is worth mentioning that in order to tune the neural network models with 6-fold cross-validation, sample size of at least 200 observational units per treatment stratum were needed to avoid violating positivity, especially as the dimension of the design matrix X increased. Thus, the results could be understood as a lower bound on sample size needed, which can be a limitation if medical studies lack many participants or the abundance of records is low. These results should be weighed against the fact that DRE-ML exhibits a higher type I error rate than its parametric counterpart even in cases where the parametric model did not match the DGP. (See supplementary material.)

4. CONCLUDING REMARKS

This paper has proposed an optimal treatment regime using the doubly robust framework with neural networks, yielding a nonparametric doubly robust estimator, DRE-ML, for observational treatment regimes. It first estimates the value function using the DRE method, then constructs the OTR. The method is more robust and computationally simpler than most existing methods, and it can handle any finite number of treatments.

The simulation results in Section 3 reveal an emerging pattern across all three treatment level types. When the outcome model's functional form was correctly specified (as in the logit-expo and multinomial-expo DGPs) DRE-Param retained a real advantage: its ATE estimates were closer to zero bias and exhibited lower variance than DRE-ML, which is expected of a model built to match the true DGP. But when the functional form was misspecified or simply unknown, as in the tanh-loggamma DGPs, this advantage reversed. DRE-ML produced estimates with lower variance and, in the three- and five-level cases, noticeably better recovery of treatment effects. The gap between DRE-ML and DRE-Param under misspecification also widened as the number of treatment levels increased, suggesting that the cost of committing to the wrong parametric outcome model compounds as the treatment space grows, while

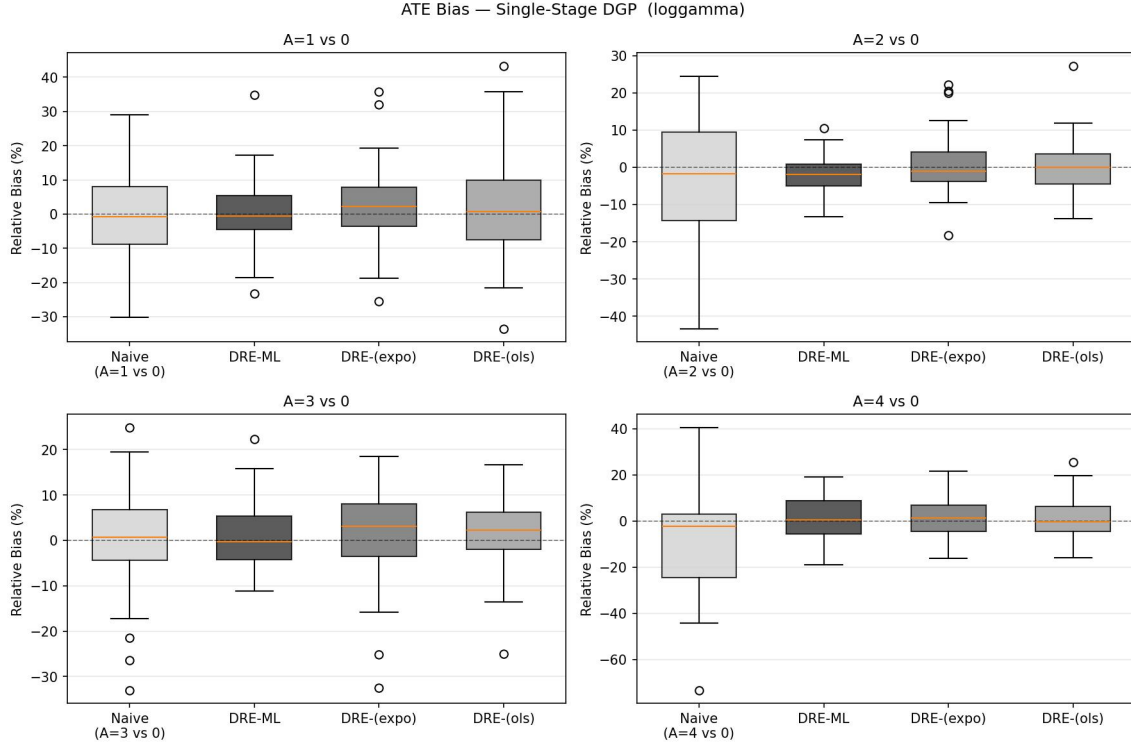


FIGURE 3.6. Difference between $\hat{\Delta}_{j,j'}$ and true $\Delta_{j,j'}$ across 30 datasets by model

DRE-ML's nonparametric flexibility offers better protection against the complexity of three- and five-treatment levels.

Further study should investigate whether the DRE-ML advantage translates into better OTRs in dynamic settings (e.g., with two- or three-stage treatment regimes). Further work could also investigate DRE-MLs performance in discontinuous or discrete outcome functions. Furthermore, the sample size requirements outlined in this study ($n = 200 \times |A_i|$) point to another direction for future work involving the lower limits of sample size, e.g., in rare diseases where a very limited number (often fewer than 100) of patients are available.

STATEMENTS AND DECLARATIONS

The authors declare that they have no conflict of interest, and the manuscript has no associated data.

APPENDIX A. SUPPLEMENTARY MATERIALS

A.1. Type I error rates. The DRE-ML estimator produces estimates for average treatment effects (ATEs) or contrasts $\hat{\Delta}_{j,j'}$. These estimates are needed to compute Optimal Treatment Regimes (OTRs) as the previous two chapters illustrated. One question that arises, then, is how precise are these estimates when the true treatment effect is zero. In other words, what is the false positive (type I error) rate when the null hypothesis is true.

This chapter conducts a simulation study with data sets where the null hypothesis is true $\Delta_{j,j'} = 0$. Since the objective is to study ATEs, and not OTRs, we limit ourselves to single-stage case study. We start with the binary treatment case, and progressive in complexity to the three-level and then five-level treatment case.

We purposely set $\delta_1 = 0$ and $\delta_2 = 0$ for each data set in

$$Y_i = g[\mathbf{x}_i^\top \boldsymbol{\beta}_Y + \delta_1 A_i + \delta_2 (A_i \times X_{p,i})] + \varepsilon_i$$

where A_i is the treatment variable, Y_i is the outcome variable, $\mathbf{x}_i$ is a vector of baseline covariates, and $X_{p,i}$ is a binary tailoring variable. $\varepsilon \sim N(0, 0.1)$.

For the three- and five-level treatment cases, we set $\boldsymbol{\delta}_1 = \mathbf{0}$ and $\boldsymbol{\delta}_2 = \mathbf{0}$.

Unlike the previous simulation chapters, we let $g(\cdot)$ be either exponential, log-gamma, scaled sigmoid, or scaled log-normal. We do that to show an additional set of cases in which the DRE-Param model matches the data-generating process (DGP), and another in which it does not.

Thus, for when the DGP is logit-lognormal, we use the scaled log-normal transformation:

$$Y_i = C \cdot \left(\sqrt{2\pi} \exp \left\{ (\mathbf{x}_i^\top \boldsymbol{\beta}_Y + \delta A_i + \delta_2 (A_i \times X_{pi})) + 0.5(\mathbf{x}_i^\top \boldsymbol{\beta}_Y + \delta A_i)^2 \right\} \right)^{-1} + \varepsilon_i$$

where $C = 10$ and $\varepsilon \sim N(0, 0.1)$.

Similarly, for when the DGP is tanh-sigmoid, we apply the following scaled sigmoid transformation:

$$Y_i = C \cdot \left(1 + \exp \left\{ \mathbf{x}_i^\top \boldsymbol{\beta}_Y + \delta A_i + \delta_2 (A_i \times X_{pi}) \right\} \right)^{-1} + \varepsilon_i$$

where $C = 10$ and $\varepsilon \sim N(0, 0.1)$.

A.1.1. Results. We begin with the binary treatment case. We generate 50 independent data sets where the null hypothesis is true ($\delta_1 = 0$ and $\delta_2 = 0$) with $n = 400$ per data set.

Within each data set, we then compute estimates $\hat{\Delta}_{0,1}$ using DRE-ML, and DRE-Param (expo) and DRE-Param (ols). For the logit-lognormal case, we also fit a logit-lognormal DRE-Param model.

Hence, Since we treatment is binary $A_i = \{0, 1\}$, we have

$$\begin{aligned} 0 &= \Delta_{j,j'} \\ &= \mathbb{E}(Y^{\{a_i=j'\}} - Y^{\{a_i=j\}}) \\ &\equiv \mathbb{E}(Y|\mathbf{x}, a = j') - \mathbb{E}(Y|\mathbf{x}, a = j) \\ &= \mu_{j'} - \mu_j \end{aligned}$$

Following the DRE procedure, for $a_i = j$, we compute unbiased estimates for μ_j

$$\hat{\mu}_j = \frac{1}{n} \sum_{i=1}^n \left(\hat{g}_j(\mathbf{x}_i) + \frac{I_{a_i=j} (y_i - \hat{g}_j(\mathbf{x}_i)) / \hat{\pi}(\mathbf{x}_i)}{n^{-1} \sum_{i=1}^n I_{a_i=j} / \hat{\pi}(\mathbf{x}_i)} \right)$$

and similarly for j' .

For DRE-ML, we approximate propensity $\pi(\cdot)$ and outcome $g(\cdot)$ functions with neural networks. (See chapter 4 and 5.)

Variance $\sigma_{\Delta_{j,j'}}$.

We pooled all 50 data sets within the DGP and computed the sample variance as:

$$\hat{\sigma}_{j,j'} = \frac{1}{(50n) - p} \sum_{i=1}^{50n} (\hat{\Delta}_{jj'}^{(i)} - \bar{\Delta}_{jj'})^2$$

TABLE A.1. Type I error rates across 50 data sets with $n = 400$ each, when the null hypothesis is true by model and DGP with binary treatment. *Signifies DGP matches DRE-param model

DGP	DRE-Type	Error rate $\hat{\Delta}_{01}$
logit-expo*	Expo	0.07
	ML	0.17
	OLS	0.01
logit-lognormal*	Lognormal	0.01
	ML	0.10
	OLS	0.03
tanh-gamma	Expo	0.07
	ML	0.20
	OLS	0.03
tanh-sigmoid	Expo	0.03
	ML	0.33
	OLS	0.03

TABLE A.2. Type I error rates across 50 data sets with $n = 600$ each, when the null hypothesis is true by model and DGP with three-level treatment. *Signifies DGP matches DRE-param model

DGP	DRE-Type	Error rate $\hat{\Delta}_{01}$	Error rate $\hat{\Delta}_{02}$
logit-expo*	Expo	0.07	0.10
	ML	0.27	0.07
	OLS	0.10	0.13
logit-lognormal*	Lognormal	0.03	0.03
	ML	0.13	0.13
	OLS	0.03	0.07
tanh-gamma	Expo	0.37	0.10
	ML	0.30	0.10
	OLS	0.43	0.13
tanh-sigmoid	Expo	0.43	0.10
	ML	0.23	0.07
	OLS	0.37	0.10

where

$$\begin{aligned}
 \bar{\Delta}_{j,j'} &= \frac{1}{50n} \sum_{i=1}^{50n} [\hat{\mu}_j(\mathbf{x}_i) - \hat{\mu}_{j'}(\mathbf{x}_i)] \\
 &= \frac{1}{50n} \sum_{i=1}^{50n} \hat{\Delta}_{j,j',(i)}
 \end{aligned}$$

Table A.1 shows the error rate for the 50 data sets by DGP and model. For the DRE-ML, we used the best neural network found in the 6-fold cross validation per data set, re-fit across the whole data set. We readily see that DRE-ML estimates have the highest type I error rate. The model seems to be too sensitive to slight differences of μ_j and $\mu_{j'}$. The parametric models do not suffer from that problem. The highest error rate is in logit-expo DGP with DRE-param (expo), and that is 0.07, which is in the upper bound of the expected rate.

TABLE A.3. Type I error rates across 50 data sets with $n = 1000$ each, when the null hypothesis is true by model and DGP with five-level treatment. * Signifies DGP matches DRE-param model

DGP	DRE-Type	Error rate $\hat{\Delta}_{01}$	Error rate $\hat{\Delta}_{02}$	Error rate $\hat{\Delta}_{03}$	Error rate $\hat{\Delta}_{04}$
logit-expo*	Expo	0.06	0.02	0.02	0.02
	ML	0.36	0.26	0.24	0.02
	OLS	0.32	0.28	0.20	0.06
logit-lognormal*	Lognormal	0.02	0.02	0.02	0.02
	ML	0.08	0.10	0.06	0.08
	OLS	0.04	0.10	0.06	0.08
tanh-gamma	Expo	0.38	0.42	0.42	0.14
	ML	0.40	0.46	0.34	0.12
	OLS	0.44	0.34	0.32	0.04
tanh-sigmoid	Expo	0.42	0.52	0.42	0.18
	ML	0.46	0.44	0.20	0.18
	OLS	0.42	0.36	0.30	0.18

Table A.2 and table A.3 show results for the three and five-level treatment respectively. The pattern that surfaces is that DRE-Param has lower type I error rates than DRE-ML, especially when DRE-Param matches the DGP with binary treatment. As we increase the treatment levels, however, type I error rates in DRE-Param are dramatically higher, reaching the 30 or even 40 percent. Only when DRE-Param matches the DGP is when the type I error rates remain below 10 percent.

This suggests that DRE-ML and DRE-Param have similar (undesirable) type I error rates when the treatment level is not binary and the DRE-Param does not match the DGP.

REFERENCES

- [1] X. Bai, A. A. Tsiatis, W. Lu, and R. Song. Optimal treatment regimes for survival endpoints using a locally-efficient doubly-robust estimator from a classification perspective. *Lifetime Data Analysis*, 23:585–604, 2017.
- [2] Y. Bengio. Learning deep architectures for AI. foundations and trends in machine learning. *Foundations and Trends in Machine Learning*, 2:1–127, 2009.
- [3] D. Blatt, S. Murphy, and J. Zhu. A-learning for approximate planning. *The Methodology Center, Pennsylvania State University*, 2004.
- [4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [5] B. Chen, A. Yuan, and J. Qin. Pool adjacent violators algorithm assisted learning with application on estimating optimal individualized treatment regimes. *Biometrics*, 78(4):1475–1488, 2022.
- [6] R. K. Crump, V. J. Hotz, G. W. Imbens, and O. A. Mitnik. Dealing with limited overlap in estimation of average treatment effects. *Biometrika*, 96:187–199, 2009.
- [7] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2:303–314, 1989.
- [8] S. Dubey, S. Singh, and B. Chaudhuri. Activation functions in deep learning: a comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.
- [9] C. Fan, W. Lu, R. Song, and Y. Zhou. Concordance-assisted learning for estimating optimal individualized treatment regimes. *Journal of the Royal Statistical Society: Series B*, 79(5):1565–1582, 2016.
- [10] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, USA, 2011. PMLR, USA.
- [11] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.
- [12] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, New York, 2nd edition, 2009.
- [13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

- [14] D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47:663–685, 1952.
- [15] J. D. Y. Kang and J. L. Schafer. Demystifying double robustness: a comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science*, 22:523–539, 2007.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. 25:1097–1105, 2012.
- [17] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, 2015.
- [18] F. Li, K. Morgan, and A. Zaslavsky. Balancing covariates via propensity score weighting. *Journal of the American Statistical Association*, 113:390–400, 2018.
- [19] J. K. Lunceford and M. Davidian. Stratification and weighting via the propensity score in estimation of causal treatment effects: a comparative study. *Statistics in Medicine*, 23(19):2937–2960, 2004.
- [20] W. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 5(3):115–133, 1943.
- [21] S. Murphy. A generalization error for q-learning. *Journal of Machine Learning Research*, 6:1073–1097, 2005.
- [22] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *Cornell University*, ArXiv:1811.03378, 2018.
- [23] M. Qian and S. Murphy. Performance guarantees for individualized treatment rules. *Annals of Statistics*, 39(2):1180–1210, 2011.
- [24] J. Robins, A. Rotnitzky, and L. Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, 89:846–866, 1994.
- [25] J. M. Robins. A new approach to causal inference in mortality studies with sustained exposure periods-application to control of the healthy worker survivor effect. *Mathematical Modelling*, 7:1393–1512, 1986.
- [26] J. M. Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics, Theory and Methods*, 23:2379–2412, 1994.
- [27] P. Rosenbaum and D. B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70:41–55, 1983.
- [28] D. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, 66:688–701, 1974.
- [29] D. Rumelhart, G. Hinton, and R. Williams. Learning representations by back-propagation errors. *Nature*, 323:533–536, 1986.
- [30] D. O. Scharfstein, A. Rotnitzky, and J. M. Robins. Adjusting for nonignorable drop-out using semiparametric nonresponse models. *Journal of the American Statistical Association*, 94:1096–1120, 1999.
- [31] S. Seaman and S. Vansteelandt. Introduction to double robust methods for incomplete data. *Statistical Science*, 33(2):184–197, 2018.
- [32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Visual Geometry Group, Department of Engineering Science, University of Oxford*, 2014.
- [33] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B*, 36:111–147, 1974.
- [34] V. Vapnik and C. Cortes. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
- [35] V. Vapnik, J. Weston, S. Barnhill, and I. Guyon. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(3):389–422, 2002.
- [36] C. J. C. H. Watkins. *Learning from delayed rewards*. PhD thesis, Cambridge University, 1989.
- [37] A. Yin, A. Yuan, and M. Tan. Highly robust causal semiparametric u-statistic with applications in biomedical studies. *International Journal of Biostatistics*, 20:69–91, 2024.
- [38] A. Yuan and M. Tan. Enhanced doubly robust procedure for causal inference. *Statistics in Biosciences*, 13:454–478, 2021.
- [39] B. Zhang, A. Tsiatis, E. Laber, and M. Davidian. A robust method for estimating optimal treatment regimes. *Biometrics*, 68:1010–1018, 2012.
- [40] J. Zhang and A. Yuan. Enhanced doubly robust estimation with concave link functions for estimands in clinical trials. *Journal of Nonparametric Statistics*, 37:841–866, 2025.