MEMENTO: Neural Model for Estimating Individual Treatment Effects for Multiple Treatments

Abhirup Mondal
mabhirup@amazon.com
Amazon
Bengaluru, India

Anirban Majumder
majumda@amazon.com
Amazon
Bengaluru, India

Vineet Chaoji
vchaoji@amazon.com
Amazon
Bengaluru, India

ABSTRACT

Learning individual level treatment effects from observational data is a problem of growing interest. For instance, inferring the effect of delivery promises on purchase of products on an e-commerce site or selecting the most effective treatment for a specific patient. Although the scenarios where we want to estimate the treatment effects in presence of multiple treatments is quite common in real life, most existing works related to individual treatment effect (ITE) are focused primarily on binary treatments and do not have a natural extension to the multi-treatment scenarios. In this paper we present MEMENTO – a methodology and a framework to estimate individual treatment effect for multi-treatment scenarios, where the treatments are discrete and finite. Our approach is based on obtaining matching representations of the confounders for the various treatment types. This is achieved through minimization of an upper bound on the sum of factual and counterfactual losses. Experiments on real and semi-synthetic datasets show that MEMENTO is able to outperform known techniques for multi-treatment scenarios by close to 10% in certain use-cases. The proposed framework has been deployed for the problem of identifying minimum order quantity of a product in Amazon in an emerging marketplace and has resulted in a 4.7% reduction in shipping costs as proved from an A/B experiment.

CCS CONCEPTS

• **Computing methodologies** → **Causal reasoning and diagnostics**; **Learning latent representations**; *Neural networks*; • **Mathematics of computing** → Causal networks.

KEYWORDS

Individual Treatment Effect, Multiple Treatment, Neural Networks, Causal Inference, Confounders

ACM Reference Format:
Abhirup Mondal, Anirban Majumder, and Vineet Chaoji. 2022. MEMENTO: Neural Model for Estimating Individual Treatment Effects for Multiple Treatments. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557125

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557125

1 INTRODUCTION

Within the causal inference literature, estimating the outcome (e.g., revenue lift) of a binary intervention (e.g., ad campaign) from historical observational data has been well studied. However, these studies [8] have primarily focused on estimating the average effect of the intervention on the outcome. The intervention (aka. *treatment*) can have heterogeneous effects on different individuals or cohorts within the population, which might not be captured in the estimation of the average treatment effect (ATE). As a result, it is imperative to estimate the treatment effect at an individual's granularity. Multiple scenarios within health care (e.g., impact of a medicine on a specific patient), public policy (e.g., impact of policies on specific cohorts), education, etc. augur for estimating the treatment effect at an individual's granularity. Additionally, estimating individual treatment effect (ITE) has the benefit of deriving other treatment effects [9].

Most techniques for causal estimation have focused on the scenario where the treatments are binary. As an extension, making predictions about causal effects with *multiple* mutually exclusive treatments, is an important problem in many domains. For instance, 1) a doctor deciding which medication results in a better outcome for a patient, 2) an e-commerce platform trying to decide on discounts, deals or offers to show to a particular customer, or 3) a teacher deciding which study program would most benefit a specific student.

Broadly, in this paper, we focus on the problem of making causal predictions based on observational data. Observational data consists of past treatments, their outcomes, and possibly more context, but without direct access to the mechanism which gave rise to the treatments. In particular, we are interested in making predictions about the outcome for the scenarios where the set of treatments are *discrete* and *finite* but take more than two unique values. A crucial aspect of inferring causal impacts from observational data is confounding. A variable which affects both the treatment and the outcome is known as a *confounder* of the effect of the treatment on the outcome. If such a confounder can be measured, the standard way to account for its effect is by "controlling" for it.

Existing literature on estimating effects of treatments is primarily focused on binary treatments and does not naturally extend to the multi-treatment scenario [12]. We present a methodology to estimate effect of treatments at an individual level where the treatments are discrete and finite. Our methodology is based on obtaining matching representations of the confounders for the various treatment types through minimization of an upper bound on

the sum of factual and counterfactual losses. We make the following contributions in this paper:

(1) We extend the methodology proposed by Shalit et. al [12] to multi treatment scenario by providing generalized definitions for errors of predictions of factuals and counterfactuals in the presence of multiple treatments and derive an upper bound to the sum of factual and counterfactual losses. The upper bound has similarities to the upper bound derived in [13] and reduces to the exact same expression in the case of binary treatments.
(2) We propose a neural model to optimize for the above loss. We call the proposed algorithm and the associated system **MEMENTO** [1], since it captures the treatments (or events) by creating representations for them. MEMENTO is a framework that provides a loss function along with a model that optimizes for the loss function. As a result, it is amenable to any underlying modeling technique that can optimization for the loss.
(3) MEMENTO is deployed in Amazon for identifying the Minimum Order Quantity (MOQ) for a product. We provide overview of the deployed production system and A/B experiments that were conducted to quantify the impact of our proposed framework over existing methodology. Additionally, to ensure reproducibility, we show and compare performances with competing algorithms on public and synthetic datasets.
(4) We propose a methodology based on uncertainty estimation to provide robust and stable estimates of the effect of each treatment in real-world scenarios.

There are a broad set of applications within Amazon where we encounter the multi-treatment setting, with the opportunity of applying MEMENTO.

(1) *Fulfillment Channel Selection*: There are multiple channels within Amazon, involving sellers and delivery carrier, to fulfill customer orders. The channels differ in terms of the fulfilling merchant, delivery speed and cost. The choice of the channel (the treatment), impacts customers' likelihood of purchasing products, which in turn impacts the revenue and the costs (the outcome).
(2) *Minimum Order Quantity for a product*: Certain products can only be purchased with a minimum order quantity (e.g., three units) constraint, due to the associated fixed costs. The treatments in this scenario are the possible values for the minimum quantity, whereas the outcome is a combination of revenue improvement and the cost savings.
(3) *Delivery Speed Optimization*: Within e-commerce, the delivery speed and its perception to customers impacts purchase decisions. The delivery date promised to the customer (taking values $\{1, 2, 3...\}$) are the mutually exclusive treatments. Correspondingly, the outcome is the purchase event or the revenue generated thereof.

MEMENTO has been launched in production from March' 21 and has been applied to the Minimum Order Quantity problem . Based on an A/B experiment conducted on an emerging marketplace, MEMENTO has an impact of 4.7% reduction in shipping costs when applied to the problem of MOQ.

[1] Anagram of the letters in the title **N**eural **Mo**del for **E**stimating Individual **T**reatment **E**ffects for **M**ultiple Treatments.

2 RELATED WORK

Much recent work in estimating causal effects revolves around the scenario where the treatment variable is a binary random variable. The two groups of populations corresponding to the two treatment types are commonly referred to as the Treatment and Control groups. Most of the methods developed for binary treatments don't have a natural extension to the scenario where the treatment is either a nominal or ordinal random variable taking more than two values [12]. For example, the popular technique of sub-classification for binary treatment has no natural extension to the multi-treatment scenario, as grouping together data points based on quantiles of the propensity score cannot be extended beyond two treatments groups. Also, the assumptions used by the techniques for binary treatment require modifications or generalizations when applied to the multi-treatment scenario. While there has been theoretical work [6, 7] to develop causal models (e.g., Generalized Propensity Score) to remove bias in scenarios with multiple treatments, practical guidance on estimating propensity scores in the multi-treatment scenario has been limited. Moreover, these are typically for estimating the ATE.

One of the most widely used approaches to estimating ATE is covariate adjustment, also known as back-door adjustment or the G-computation formula [15, 16]. In its basic version, for binary treatment, covariate adjustment amounts to estimating the functions $m_1(x)$, $m_0(x)$ ($m_i(x)$ is the conditional expectation of the outcome given input x under treatment i). Therefore, covariate adjustment methods are the most natural candidates for estimating Individual Treatment Effects as well as Average Treatment Effects, using the estimates of $m_t(x)$. This class of methods also has a natural extension to the multi-treatment scenario, where we estimate the set of conditional mean functions $m_i(x), i \in \{1, 2, ..., K\}$ where K is the number of treatments.

Another widely used family of statistical methods used in causal effect inference are weighting methods. Methods such as propensity score weighting [1] for binary treatments, perform re-weighting of the units in the observational data so as to make the treated and control populations more comparable. These methods are naturally designed to obtain population level estimates such as ATE. Nonetheless they can be modified to estimate an individual level effect. While the individual level estimate obtained from methods such as propensity score weighting provide unbiased and consistent estimate of Individual Treatment Effect, they have quite high variance. Doubly robust methods combine re-weighting the samples and covariate adjustment in clever ways to reduce model bias [2], but suffer from the problem of even higher variance for individual level estimates at the benefit of providing lower bias. Multiple efforts exist for propensity score estimation and weighting [10, 11, 14] within the multi-treatment setup.

Finally, our paper builds on work by Shalit et al. [20], where they provide two algorithms called Counterfactual Regression (CFR) and Treatment Agnostic Representation Network (TARNet) to obtain individual level treatment effect estimates for binary treatment. In this work the authors provide an upper bound to the estimate of

the Precision in Estimation of Heterogeneous Effect (PEHE) and minimize this upper bound to obtain matching representations for the two treatment groups. The distance between the Treatment and Control populations is captured through a proper Integral Probability Metric (IPM) [21]. There have been other efforts [19] to propose matching algorithms (e.g., vector matching) within the multi-treatment realm.

Recently, there have been multiple papers [17, 18] that apply deep generative models for causal inference. In [17], authors utilize task specific embeddings to scale to multiple treatments, even in the scenario when a treatment is a subset of treatments. [23] proposes a similarity preserved individual treatment effect estimation method (SITE) based on deep representation learning. The paper proposes an approach to balancing the distribution, keeping in mind the local similarity between units in the observational data. Yoon et al. [24] propose an approach (GANITE) based on Generative Adversarial Network (GAN) to estimate the ITE in the multi-treatment scenario.

3 ESTIMATING ITE FOR MULTIPLE TREATMENTS

We present an algorithm to estimate individual treatment effect (ITE) on a problem domain with multiple treatments. We provide a bound on the expected error in estimating ITE in terms of (a) generalization error in learning *factual* outcome and (b) an Integral Probability Metric (IPM) defined over the distribution of pairs of treated units.

3.1 Background and Notations

We denote the set of potential treatments or interventions by $\mathcal{T}$. Note that in our setting, $|\mathcal{T}| = K \geq 2$. Let $\mathcal{X} \subset \mathbb{R}^d$ be the set of feature vectors (referred to as confounders) used to represent individual datapoints and $\mathcal{Y} \subset \mathbb{R}$ be the set of potential outcomes. Given a datapoint $x \in \mathcal{X}$, let $y_t(x) \in \mathcal{Y}$ be the potential outcome of applying the treatment $t \in \mathcal{T}$.

Suppose we have access to n independent and identically distributed observations $i = 1, 2, \cdots n$ where each observation is of the form $(x^{(i)}, y^{(i)}, t^{(i)})$ representing feature vector $x^{(i)} \in \mathcal{X}$, treatment $t^{(i)} \in \mathcal{T}$ and potential outcomes $y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \cdots y_K^{(i)}) \in \mathcal{Y}^K$. Note that only the outcome $y_{t_i}^{(i)}$, corresponding to the treatment being applied, is observed whereas all other outcomes are *counterfactual*. Given this setup, the Individual Treatment Effect (ITE) of sample x with respect to treatments $t, t' \in \mathcal{T}$ is defined as the following quantity:

$$\tau_{t,t'}(x) = \mathbb{E}\left[y_t - y_{t'} \mid x\right] \tag{1}$$

The fundamental problem in causal estimation is that we can observe only one of the outcomes $y_1, y_2, \cdots y_K$ depending on which treatment is applied and the others are never observed. Therefore unlike supervised learning set-up, we can not train a machine learning model to estimate $\tau_{t,t'}$ directly. To make estimation feasible, we need certain assumptions on the data generating process. The joint distribution of the outcome, confounders and treatment random variables is governed by the graphical model shown in Figure 1. Let $p(x, y_{1\cdots K}, t)$ be the joint distribution. The standard practice is to assume that the potential outcomes $y_{1\cdots K}$ are independent of the treatment variable t conditioned on the confounders x, i.e.,

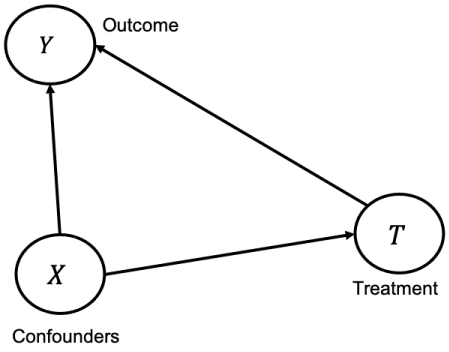


Figure 1: The Graphical Model representing the joint distribution of confounders (X), treatment(T) and potential outcomes(Y).

$$y_{1\cdots K} \perp\!\!\!\perp t \mid x$$

This is known as *strong ignorability* [16] which implies that all confounding variables are accounted for our definition of the feature space $\mathcal{X}$. Under strong ignorability, we can treat nearby datapoints in $\mathcal{X}$-space as having come from a fully randomized experiment.

Given an estimate $\hat{\tau}_{t,t'}(x)$ of the ITE, the *Expected Precision in Estimation of Heterogeneous Effect* (PEHE [5]) loss is defined as

$$PEHE_{t,t'} = \int_{\mathcal{X}} \left(\hat{\tau}_{t,t'}(x) - \tau_{t,t'}(x)\right)^2 p(x)\, dx \tag{2}$$

For binary treatment settings, Hill et al. [5] have posed causal inference estimation as a machine learning problem of minimizing the PEHE loss. However, there is no natural extension of PEHE to the multi-treatment setup. We circumvent this difficulty by using a result from Shalit et al. [20] which provides an upper bound to PEHE using a sum of factual and counterfactual losses under certain assumptions. The sum of factual and counterfactual losses have a natural extension to the multi-treatment scenario. We derive an algorithm to minimize this sum, which provides us with estimates of the treatment effects.

3.2 Multi-Treatment Loss Functions

We assume predictions are performed using a *representation mapping* of the form $\phi : \mathcal{X} \to \mathcal{Z}$ where $\mathcal{Z}$ is a latent representation space for each observation. Further assume that ϕ is twice differentiable and one-to-one function [20] for reasons to be explained later.

Let $h : \mathcal{Z} \times \mathcal{T} \to \mathcal{Y}$ be a hypothesis that takes the pair (x, t) as input and estimates the potential outcome $y_t(x)$. Associated with this prediction tasks is a pre-determined loss function (e.g., squared error loss as used in PEHE) $\mathcal{L}(y, h(\phi(x), t))$ and hence an expected loss for the input (x, t):

$$l(x, t) = \int_{\mathcal{Y}} \mathcal{L}(y_t, h(\phi(x), t))\; p(y_t|x)\; dy_t \tag{3}$$

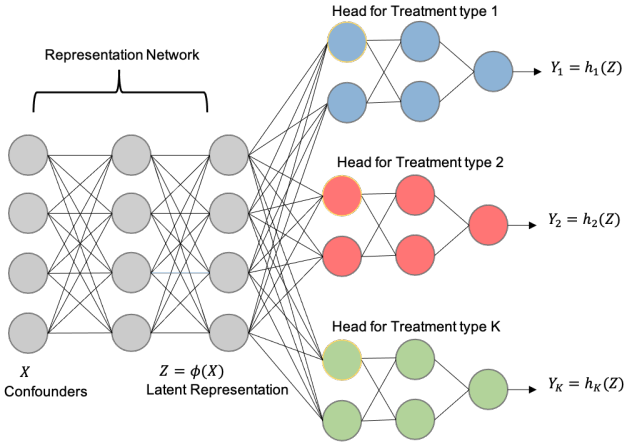


Figure 2: Multi-headed network for minimizing the upper bound. The pairwise distance between the distributions of the treatment groups is calculated as the MMD of the output vectors in the representation network. The factual loss is calculated using the corresponding heads in the outcome network.

Finally, the expected loss incurred by our hypothesis h is given by

$$\mathscr{L}(h) = \int_{\mathscr{X}} \int_{\mathscr{T}} l(x,t)p(t)\ p(x \mid t)\ dx\ dt \tag{4}$$

Given a dataset $\mathscr{D} = \{(x_i, y_i, t_i)\}_{1\cdots n}$ of observations, causal estimation of treatment effects involves two prediction tasks:

- **Factual predictions:** For the observed treatment t, we predict $y_t(x)$. This leads to *factual* loss as defined below. Note that this task is same as standard supervised ML models.

$$\mathscr{L}_F(h) = \sum_t p(t) \int_{\mathscr{X}} l(x,t)\ p(x|t)\ dx$$

- **Counterfactual predictions:** For each unobserved treatment t', we predict potential outcome $y_{t'}(x)$. This gives rise to *counterfactual* loss which is specific to causal estimation tasks:

$$\mathscr{L}_{CF}(h) = \sum_t \sum_{t' \neq t} p(t) \int_{\mathscr{X}} l(x,t')\ p(x|t)\ dx$$

Note that we cannot directly estimate $\mathscr{L}_{CF}(h)$ using only the observed data. Instead we derive an upper bound on $\mathscr{L}_F(h) + \mathscr{L}_{CF}(h)$ i.e., the sum of the factual and counterfactual losses. We show that this upper bound can be estimated efficiently using training data at hand and we present an algorithm to minimize it.

3.3 Upper Bounding the Loss Function

The following theorem provides an upper bound to the sum of factual loss and counterfactual loss:

Theorem 3.1. *Let $\Phi : \mathscr{X} \to \mathscr{Z}$ is a twice differentiable and one-to-one function. Let $h : \mathscr{Z} \times \mathscr{T}$ be a hypothesis and $\mathscr{G}$ be a family of loss functions. Further define $u_t = p(t)$. There exist a constant $C > 0$ such that the total loss is bounded as,*

$$\mathscr{L}_F(h) + \mathscr{L}_{CF}(h) \leq \sum_t \int_{\mathscr{X}} l(x,t)p(x \mid t)\ dx + C \cdot \sum_t \sum_{t'} (u_t + u_{t'})\ IPM_{\mathscr{G}}(p(x \mid t), p(x \mid t')) \tag{5}$$

Proof. Define $u_t = p(t)$, the marginal probability of observing the treatment t. For two treatment t and t', consider

$$\begin{aligned} & u_t \int_{\mathscr{X}} l(x,t)p(x|t)dx - u_t \int_{\mathscr{X}} l(x,t')p(x|t')dx \\ &= \int_{\mathscr{X}} u_t l(x,t)(p(x|t) - p(x|t'))dx \\ &\leq u_t.C.IPM((p(x|t), p(x|t')) \end{aligned} \tag{6}$$

, where where IPM [8] is the Integral Probability Metric. Integral Probability Metric (IPM) is a class of metrics over probability distributions [4, 21, 22]. For two probability distributions p, q defined over $\mathscr{S} \subseteq \mathbb{R}^d$ and a family of functions $G : \mathscr{S} \to \mathbb{R}$, IPM is defined as

$$IPM_G(p,q) = \sup_{g \in G} \left| \int_{\mathscr{S}} g(s)\ (p(s) - q(s))\ ds \right| \tag{7}$$

IPM is a true distance metric for probability distributions as it satisfies all the three properties: (a) $IPM_G(p,q) \geq 0$ and the equality is achieved only when $p = q$, (b) symmetric and (c) satisfies triangle inequality i.e., $IPM_G(p,q) \leq IPM_G(p,r) + IPM_G(r,q)$. C is a positive constant dependent on the class of witness functions corresponding to the IPM.

Summing both sides over $\sum_t \sum_{t' \neq t}$, we see that the LHS reduces as follows:

$$\sum_t \sum_{t' \neq t} u_t \int_{\mathscr{X}} l(x,t)p(x|t)\ dx - \sum_t \sum_{t' \neq t} u_t \int_{\mathscr{X}} l(x,t)p(x|t')\ dx \tag{8}$$

We identify the first term as counter-factual loss $\mathscr{L}_{CF}(h)$.
For the second term, we see that interchanging the order of summation leads to a weighted version of the factual loss:

$$\begin{aligned} & \sum_t \sum_{t' \neq t} u_t \int_{\mathscr{X}} l(x,t)p(x|t')dx \\ &= \sum_{t'} \sum_{t' \neq t} u_t \int_{\mathscr{X}} l(x,t)p(x|t')dx \\ &= \sum_{t'} \left\{ \sum_{t \neq t'} u_t \right\} \int_{\mathscr{X}} l(x,t)p(x|t')dx \\ &= \sum_{t'} \{1 - u_{t'}\} \int_{\mathscr{X}} l(x,t)p(x|t')dx \\ &= \sum_{t'} \int_{\mathscr{X}} l(x,t)p(x|t)\ dx - \mathscr{L}_F(h) \end{aligned} \tag{9}$$

Algorithm 1 Algorithm for minimizing upper bound and estimating outcomes for each treatment type

Input: Observed data points $\{(y_1, x_1, t_1), (y_2, x_2, t_2), ...(y_n, x_n, t_n)\}$, Loss function $\boldsymbol{l(y(t), x, t)}$, Representation network architecture with initial weights $\boldsymbol{W}$, Outcome network architecture with initial weights $\boldsymbol{V}$, Kernel for calculating sample estimate of Maximum Mean Discrepancy $\boldsymbol{K(.,.)}$, Imbalance penalty hyperparameter $\boldsymbol{\alpha}$, Learning rate η for SGD.
Compute marginal probabilities for each treatment type $u_i = P(t_i = 1)$ for $i \in \{1, 2, ..., K\}$
while not converged **do**
  Sample minibatch $\{i_1, i_2, i_3, ..., i_m\} \subset 1, 2, 3, ..., n$
  Calculate weighted factual loss for the sample using the expression in first term in Equation 5.
  For $i \neq j$, calculate the sample estimator of MMD following Equation 10 for all the pairwise distributions $MMD(P_i, P_j)$ and calculate the weighted sum of the pairwise MMD estimates where the weights are given by $(u_i + u_j)$.
  Calculate gradient of weighted factual loss obtained in step 5. w.r.t. $\boldsymbol{W}$ and $\boldsymbol{V}$, say $\boldsymbol{g_1}$ and $\boldsymbol{g_2}$.
  Calculate gradient of the weighted sum of sample estimates of MMD obtained in step 6. w.r.t. $\boldsymbol{W}$, say $\boldsymbol{g3}$
  $\boldsymbol{[W, V]} \leftarrow [\boldsymbol{W} - \eta(g1 + \alpha g_3), \boldsymbol{V} - \eta g2]$
  check convergence criteria
end while

Rearranging the terms, we obtain,

$$\mathscr{L}_F(h) + \mathscr{L}_{CF}(h) \leq \sum_t \int_{\mathscr{X}} l(x,t)p(x \mid t)\, dx + C \cdot \sum_t \sum_{t'} (u_t + u_{t'})\, IPM(p(x \mid t), p(x \mid t'))$$

□

The first term in the RHS of Equation 5 is weighted factual loss, where the weights are $1/u_t$. the second term in the RHS attempts to match the distribution of the confounding variables under different treatments. Note that frequent treatment pairs are given higher importance in the matching criteria through the multiplier $u_t + u_{t'}$.

The second term can be estimated depending on the class of functions $\mathscr{G}$ used in the calculation of IPM. If the loss function belongs to the unit ball in a reproducing kernel Hilbert space (RKHS), then the IPM reduces to Maximum Mean Discrepancy (MMD [4]) measure between two distributions. The following lemma from Gretton et al. [4] provides an efficient estimate of MMD from a sample

Lemma 3.2. *Let $x_1, x_2, ..., x_m$ and $x'_1, x'_2, ..., x'_n$ are the two sets of samples from the distributions p and q respectively. Given kernel K, the following is a consistent and unbiased estimator of MMD [4]:*

$$\widehat{MMD}(p,q) = \frac{1}{m(m-1)} \sum_{1 \leq i \neq j \leq m} K(x_i, x_j) + \frac{1}{n(n-1)} \sum_{1 \leq i \neq j \leq n} K(x'_i, x'_j) - \frac{2}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} K(x_i, x'_j) \tag{10}$$

Although this estimator requires $O(n^2)$ in the number of samples, Gretton et al [4] outlines how one can obtain a linear time estimator albeit higher variance. The upper bound on the loss function (Equation 5) can be minimized using any optimization technique.

3.4 The MEMENTO Algorithm

In this section, we present MEMENTO, a neural network to estimate the causal impact for multiple treatments. MEMENTO employs an end-to-end minimization procedure that simultaneously meets two goals: (a) learn latent representation of datapoints to minimize imbalance between the treatment groups (b) minimize prediction error on factual outcomes.

MEMENTO achieves these objectives in two stages. In the first stage, it learns a representation $\phi : \mathscr{X} \rightarrow \mathscr{Z}$ to transform an input feature vector $x \in \mathscr{X}$ to a latent representation $\phi(x) \in \mathscr{Z}$. Since the treatment groups are generally imbalanced in the $\mathscr{X}$-space, the goal is to improve covariate balance in the $\mathscr{Z}$ space through the mapping ϕ. In MEMENTO we use three-layer fully connected networks with tanh non-linearity to represent ϕ. The parameters are learned by minimizing $MMD(p(\phi(x) \mid t), p(\phi(x) \mid t'))$ for all pair of treatments (t, t').

In the second stage, MEMENTO uses learned representations $\phi(x)$ to predict the outcome variable y. We use separate *heads* to parameterize each treatment outcome. More specifically, we use a parametric function $h_t(\phi(x))$ to estimate the effect of treatment t on input x. The function h_t is represented as a deep neural network with parameter set θ_t. With this approach, we gain statistical power through a common representation framework (ϕ) while retaining treatment-specific variation through the output networks $\{h_t | t \in \mathscr{T}\}$.

It is important to note that both the representation and output layers are trained jointly. As mentioned in Theorem 3.1 and Lemma 3.2, the network is trained by minimizing the following upper bound on the loss function,

$$\frac{1}{N} \sum_{i=1}^{N} \frac{1}{u_t} \mathscr{L}(y_i, h_{t_i}(\phi(x_i))) + R(\phi, h_{1...T}) \quad \alpha \cdot \sum_{t \neq t'} MMD(p(\phi(x) \mid t), p(\phi(x) \mid t')) \tag{11}$$

Here $R(\cdot)$ is a regularizer term to control the model complexity. The model is trained using stochastic gradient descent and the prediction error is backpropagated through both the representation and outcome networks to update their respective set of parameters. The exact form of the constant α in Equation 11, for a given loss

function is generally unknown and we treat it as a hyper-parameter for MEMENTO.

3.5 Uncertainty Estimates

The neural model proposed in the previous section provides us with point estimates of the outcome variable given the set of Confounders and the Treatment variable. Specifically, the model produces the counterfactual point estimates of $E(y|x, T = t')$ but does not provide any information about the uncertainty of this estimate. However, in real life scenarios, it is prudent that we have some sense of the reliability of this estimate before we select and apply the best treatment as suggested by the neural model. To address this problem, we provide a methodology to obtain uncertainty estimates in the form of $Var(y|x, T = t')$ in addition to the already existing point estimates of $E(y|x, T = t')$. Using the uncertainty estimate in the form of $Var(y|x, T = t')$, we can prune the treatments which have high uncertainty (e.g., $Var(y|x, T = t') \geq \tau$) and use only the rest of the point estimates for downstream tasks.

To obtain the uncertainty estimates, we adopt the methodology in [3]. We introduce dropout layers in both the representation network and outcome network. Since, the loss function minimization is agnostic to the structure of the network, the introduction of dropout layers does not cause any issues with the minimization of the upper bound. Having trained the network in the presence of dropout layers, in the estimation period we freeze the network parameters and perform multiple forward passes. During the forward passes, we let the dropout layers to be active. Thus, for the i^{th} forward pass, we obtain $\hat{y_i}(x, T = t')$ for every treatment t'. Finally, to obtain the epistemic uncertainty estimates, we calculate for every treatment t', the following estimate of the variance:

$$Var(\hat{y}|x, T = t') = \frac{1}{n} \sum_{i=1}^{i=n} (y_i(x, \hat{T} = t') - \bar{y}(x, T = t'))^2$$

We can use this variance estimate to prune the treatments which have high uncertainty ($\{t' : Var(\hat{y}|x, T = t') \geq \tau\}$ for a pre-specified τ) around their point estimates. We demonstrate in the Minimum Order Quantity problem, how the uncertainty estimates are used to recommend robust treatments.

4 EXPERIMENTS AND RESULTS

We present results on synthetic datasets and real-world datasets. A challenge w.r.t. evaluating counterfactual prediction tasks is the lack of ground truth data except for Randomized Control Trials (RCT). But most real life observational data is not akin to RCTs and mostly are observational data. Due to lack of proper ground truth data, we either simulate the outcome variable (semi-synthetic data) or use appropriate heuristics to get approximate ground truth data. We use cross-entropy and RMSE as the choice of loss functions for categorical and continuous outcome types, respectively. The choice of the kernel for estimating MMD is the Squared Exponential Kernel. We compare our methodology with four baseline techniques: NN1: using a single DNN model and treatment as a feature, NN2: using separate DNN models for each treatment type, IPW: Inverse propensity estimate, DR: Doubly robust estimate, TARNet: We just use our network architecture but don't optimize using the loss in Equation 11.

4.1 Minimum Order Quantity

To sustainably fulfill shipments of very low priced products, many e-commerce platforms set a lower bound viz. Minimum Order Quantity (MOQ) on the number of units of the product that can be purchased at a time. For example, a MOQ of 3 on a soap would mean that the customer has to purchase at least three units of that soap. While MOQ reduces shipping cost via shipping multiple units together, it may negatively impact purchase decisions if placed on incorrect products. For example, setting MOQ of 3 on a mobile charger would lead to poor customer experience and subsequently would lead to drop in sales of the product. Thus, we would need to optimally select the products on which MOQ can be applied and also select the value of MOQ for the product. In this context, we apply MEMENTO to determine the impact of MOQ (considered as Treatment) on the purchase decisions (considered as outcome variable). The Treatment variable takes values 1, 2, 3, ..., k and the outcome variable(*conversion*) is a $0-1$ indicator random variable corresponding to whether a purchase happens for a product page view or not. To formulate the MOQ selection as an optimization problem, we introduce the following notations and definitions:

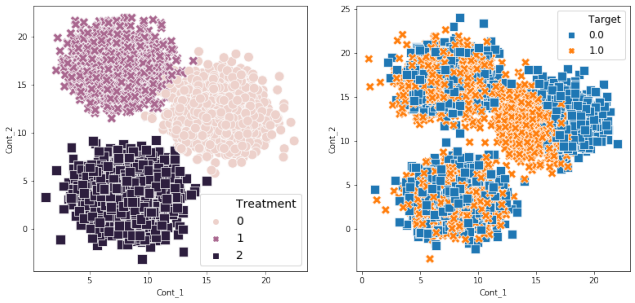
(1) Define $S_i(k)$ as the shipping cost of shipping k units of the i^{th} product together. We assume that $S_i(k) \leq k * S_i(1)$ for every product, since shipping multiple units together is less costlier than shipping each unit individually,
(2) f_{ik} as 0, 1 valued indicator variable indicating whether the i^{th} product has been applied the MOQ of k,
(3) c_{ik} is the estimate of conversion for the i^{th} product when applied the Treatment (MOQ) of k. The conversion estimates for different MOQ values are obtained using MEMENTO,
(4) p_i is the price of the i^{th} product.

We can pose the MOQ selection problem as trying to find the MOQ values which leads to maximum reduction in shipping cost but does not reduce revenue beyond a pre-specified threshold (R).

$$\begin{aligned} &argmin_{f_{ik}} \sum_i S_i(k) \cdot c_{ik} \cdot f_{ik} \\ &s.t. \sum_i k \cdot f_{ik} \cdot c_{ik} \cdot p_i \geq R \end{aligned} \tag{12}$$

To solve the the above, we would plug in the estimates of c_{ik} obtained using MEMENTO. and use any LP solver to obtain the MOQ recommendations for each product.

To obtain the estimates of Conversion for each product page view as a function of MOQ, we train the model using historical data on MOQs offered and conversion. Historically MOQ was applied if the product had a high repeat purchase rate. For example, if the product was observed to be purchased by customers repeatedly over a fixed period of time (say 3-4 months), then those products were put up for MOQ. A higher proportion of repeat purchases would lead to the MOQ being set to a higher value. Along with the repeat purchase signal, other features like product price, product category etc. were used to devise a rule based system to assign MOQ on products. This mechanism of setting MOQ on products led to the presence of selection bias in the training data. As such training supervised Machine Learning Models on such a dataset would lead to incorrect inferences about the Treatment Effect. We use MEMENTOto train the neural model on this dataset. Thus, we



(a) Treatment Allocation with selection bias.

(b) Natural Outcome for each datapoint.

Figure 3: Treatment allocation and outcome distribution for the synthetic dataset.

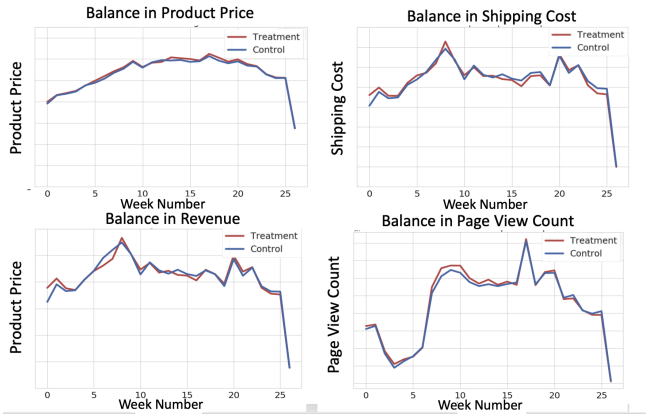


Figure 4: Balance during pre-experiment period

estimate c_{ik} as: $P(y_i|x_i, T = k)$, where T is the MOQ for the product and y is the conversion indicator and x consists of the context features. The context features comprise of the product features (like price, category, repeat purchase rate etc.), seller features (like tenure, fulfillment channel etc.), customer features (like Prime Membership, tenure etc.) and product page level features (like promised delivery speed, deals/discounts etc.).

4.1.1 Online A/B Experiment. To test the improvement obtained using MEMENTO,we performed an A/B experiment for 3 months in 2021 in an emerging marketplace in Amazon. In the A/B experiment, we compared the MOQ recommendations obtained from MEMENTOwith the recommendations generated by the incumbent mechanism.

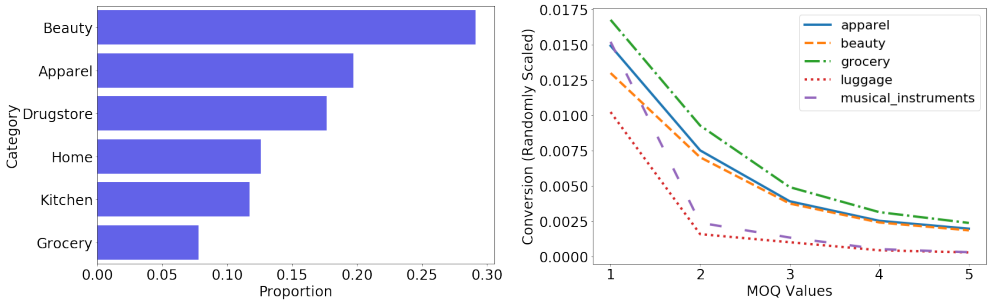
To initiate the A/B experiment, we divided the set of products into two homogenous groups of Control and Treatment such that the key factors like shipping cost, revenue, price, category etc. of the products in the Control and Treatment were very similar over a baseline period of 26 weeks. Refer to Figure 4 to see the balance in the key factors in the pre-experiment period. During the experiment, the Control group of products had their MOQ set according to the incumbent mechanism and the Treatment group of products had their MOQ set according to the solution of equation 12 with the conversion estimates being obtained from MEMENTO. To measure the success of the A/B experiment, we measured 'shipping cost per product per week' and 'revenue per product per week' for all the products in both Control and Treatment. At the conclusion of the A/B experiment we saw a 4.7% reduction in shipping cost in Treatment as compared with Control. To check for the statistical significance of the observed impact, we performed the following hypothesis test :

$$H_0 : \mu_0 \leq \mu_1 \tag{13}$$

$$H_1 : \mu_0 > \mu_1 \tag{14}$$

where μ_0 and μ_1 are the means of the Control and Treatment population. The null hypothesis corresponds to the case where the MOQ recommendations does not lead to a reduction in the shipping cost, whereas the alternate hypothesis corresponds to the case where the MOQ recommendations lead to a reduction in the shipping cost. When we performed the test with the metric of shipping cost per product per week, we were able to reject the null-hypothesis (p-value ≤ 0.01). To understand the impact on revenue, we performed a hypothesis test on the equality of means of the revenue of the Control and Treatment groups. For revenue, we used the metric of revenue per product per week and the null hypothesis was accepted (p-value = 0.8904) in this case. Thus, through the experiment, we were able to conclude that using our framework, we were able to obtain statistically signification reduction in shipping cost while not causing any significant reduction in revenue.

4.1.2 Production System. MEMENTO was launched in production since 2021 and has been used to recommend Minimum Order Quantity of products in an emerging marketplace. Figure 6 shows the high-level architecture diagram of the production system. The customer, seller, product and view level attributes are pulled from backend data sources (s3/Redshift cluster) to generate features using Apache Spark and scored through the Model on a cloud instance. After the model scoring, an optimization routine is run to compute the recommended value of MOQ for every product. The MOQ recommendations from the system are then consumed by the downstream systems for surfacing them to the customers at the time of product page view. The MOQ value shown and conversion data is fed back to the database for model retraining.



(a) Proportion of products in each category put up for MOQ recommendations.

(b) Drop in conversion due to MOQ across different categories of products.

Figure 5: Minimum order quantity data-set.

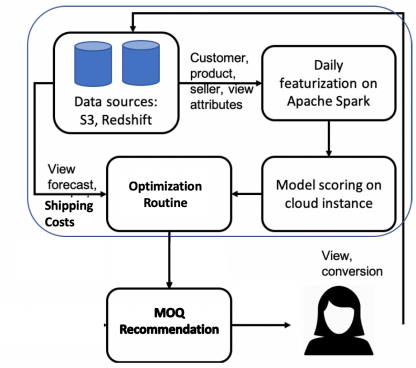


Figure 6: High Level Production System Architecture

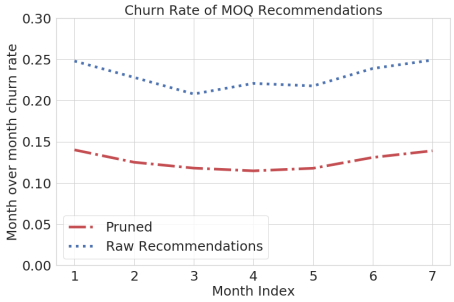


Figure 7: Churn Rate (scaled) reduction using uncertainty estimates

To understand the qualitative aspect of the recommendations, we also look into the way Conversion changes for different categories due to setting different MOQ values. We see from Figure 5(b) that for certain categories (like Luggage) the drop in conversion is very sharp while for other categories (like Grocery, Beauty etc.) the drop in conversion is more steady. This observation is in accordance with our intuition that categories consisting of consumables like Grocery, Beauty etc. are more suited for being put up for MOQ.

4.1.3 Robustness and Stability of MOQ Recommendations. To add robustness to the recommendations and to protect customer experience, we also use the uncertainty estimates obtained from the MEMENTO to prune estimates with high variance. If $Var(c_{ik})$ is higher than a specified threshold τ, then for that product those treatments are not included in the optimization routine. Specifically, we set $f_{ik} = 0$ if $Var(c_{ik}) \geq \tau$. To find out an optimal value of τ, we measure the churn of MOQ recommendations over time. Intuitively, if there is a frequent change in the MOQ recommendations of a product (due to unreliable estimates from MEMENTO), then it would lead to a confusing and poor experience for the customers. We define churn rate as the proportion of products in the current time period which different MOQ recommendation that was present in the previous time period. We observe from offline analysis that the pruning using threshold on variance significantly reduces churn in the MOQ recommendations of the products. We compare two strategies for this purpose, one where we prune the recommendations using the variance estimates and the other where we don't use any pruning. We generate the offline recommendations for a period of 8 months and observe the churn rate on a month-over-month basis. We observe from Figure 7, that the pruning using uncertainty estimates leads to significant reduction in churn of the recommendations.

4.2 Synthetic Data

We develop further intuition into the workings of our methodology by studying its behavior on synthetic data. We simulate data according to the following generative process:

(1) Confounders (X): We generate a set of continuous random variables using Scipy's make_blobs [2] function. We create three blobs corresponding to three treatment types. The standard deviation of the blobs are chosen such that they don't have much overlap with each other.
(2) Treatments (t): Three treatments (called 0, 1, 2) are assigned to the three blobs with some noise added to the assignment process. We demarcate the blobs as $blob_0, blob_1, blob_2$ and

[2] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_blobs.html

associate the treatments 0, 1, 2 with them respectively. A datapoint in $blob_i$ is assigned to treatment i with a probability of 0.95 and to treatment j (where $j \neq i$) with probability 0.025. The allocation can be visualized in Figure 3(a). This assignment process ensures that the type of treatment a datapoint receives is a direct function of the Confounders (X).

(3) Outcome (y): The outcome variable depends on the confounders and treatment according to the following functions: $(y|x, t = 0) \sim (0.1 \cdot f(X) - 0.5 \cdot g(X))^\intercal \cdot \mathbf{1} + \epsilon_0$, where f and g are polynomials of order 2 and 3

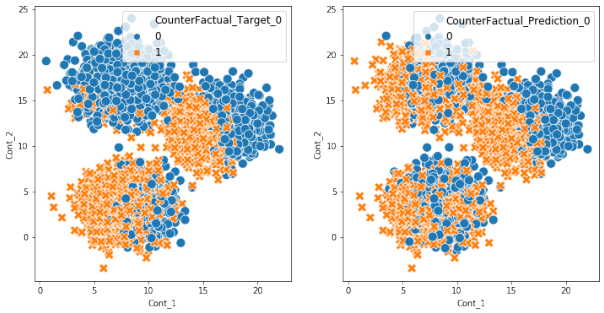
$(y|x, t = 1) \sim cos(2 * (t) * X)^\intercal \cdot \mathbf{1} + \epsilon_1$

$(y|x, t = 2) \sim sin(t * (X \cdot R))^\intercal \cdot \mathbf{1} + \epsilon_2$, where R is a random square matrix
, where $\epsilon_0, \epsilon_1, \epsilon_2$ correspond to homoscedastic errors. Finally we apply the transform $y = 0$, whenever $y \leq thresh$, else $y = 1$ to convert y to a binary random variable. The distribution of the output variable can be visualized in Figure 3(b).

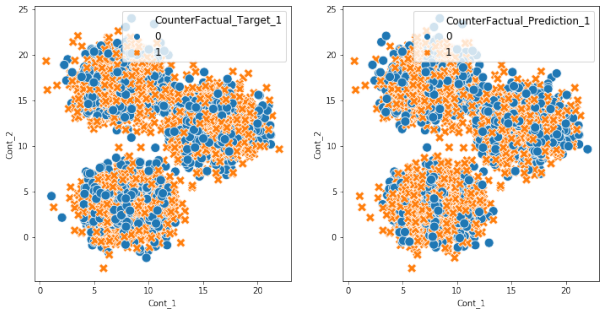
Method	ROC-AUC	Improvement(in %)
NN1	0.8134	10.499%
NN2	0.832	8.029%
IPW	0.8193	9.703%
DR	0.8014	12.154%
TARNET	0.8701	3.298%
MEMENTO	**0.8988**	NA

Table 1: ROC-AUC of counterfactual prediction on the synthetic dataset.

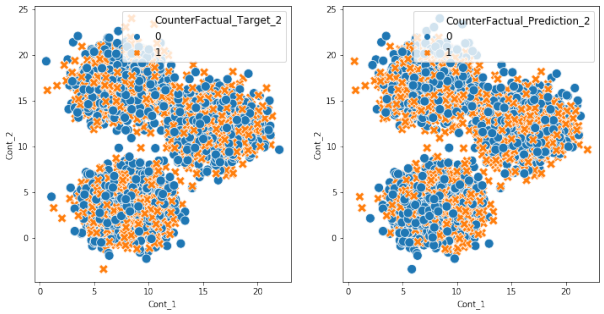
The natural outcome for every treatment type and the selection bisa present in the generated data can be observed in Figure 3. We train our model using the generated data as training data and try to predict the counterfactuals for every possible treatment value for every datapoint. We observe from Figure 8 that we are able to predict the counterfactual for each treatment type well and are able to handle the selection bias present in the training data. To quantitatively evaluate the performance of the counterfactual predictions, we predict for each datapoint, the counterfactual outcomes corresponding to all the three treatment types. We also have access to the ground truth value of the outcome for each of the three treatment types using data generative process described above. Since, the outcome is 0-1 valued, we used ROC-AUC as the choice of metric to evaluate the performance of the counterfactual predictions. In Table 1, we show the performance of our algorithm in comparison with several baselines in predicting the counterfactual outcome. As can be seen, MEMENTO achieves significant improvement over supervised baselines (NN1, NN2) as well as improves over the IPW and DR estimates. Finally MEMENTO achieves better results than just using the TARNET architecture showing the importance of our proposed loss based on the upper bound for factual and counterfactual losses.



(a) Predictions when all points forced through treatment $t = 0$



(b) Predictions when all points forced through treatment $t = 1$



(c) Predictions when all points forced through treatment $t = 2$

Figure 8: Comparison between factual (left plots) and counterfactual (right plots) predictions on the synthetic dataset. For all the treatment types we are able to predict both the factual and counterfactual patterns of the outcome variable.

5 CONCLUSION

In this paper we presented a methodology to get individual level counterfactual estimates in the presence of multiple treatments. We demonstrated through extensive experiments on Amazon and Public data the superiority of our proposed method over existing popular techniques. This methodology would have an extremely wide applicability for various businesses across Amazon and externally as well.

Some of the future research aspects could be directed towards providing robust estimates even in the presence of missing confounders and extending to continuous treatments.

REFERENCES

[1] Peter C Austin. 2011. An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies. *Multivariate Behavioral Research* 46, 3 (2011), 399–424.

[2] Michele Jonsson Funk, Daniel Westreich, Chris Wiesen, Til Stürmer, M. Alan Brookhart, and Marie Davidian. 2011. Doubly Robust Estimation of Causal Effects. *American Journal of Epidemiology* 173, 7 (03 2011), 761–767.

[3] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. arXiv:1506.02142 [stat.ML]

[4] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A Kernel Two-Sample Test. *J. Mach. Learn. Res.* 13 (2012), 723–773.

[5] Jennifer L. Hill. 2011. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics* 20, 1 (March 2011), 217–240. https://doi.org/10.1198/jcgs.2010.08162

[6] Kosuke Imai and David A. Van Dyk. 2004. Causal Inference with General Treatment Regimes: Generalizing the Propensity Score. *J. Amer. Statist. Assoc.* 99, 467 (2004), 854–866.

[7] Guido W. Imbens. 2000. The Role of the Propensity Score in Estimating Dose-Response Functions. *Biometrika* 87, 3 (2000), 706–710.

[8] Guido W. Imbens. 2004. Nonparametric estimation of average treatment effects under exogeneity: a review. *REVIEW OF ECONOMICS AND STATISTICS* (2004), 4–29.

[9] Patrick Kenneth Lam. 2013. *Estimating individual causal effects.* Ph.D. Dissertation.

[10] Fan Li and Fan Li. 2019. Propensity score weighting for causal inference with multiple treatments. *Ann. Appl. Stat.* 13, 4 (12 2019), 2389–2415.

[11] Ariel Linden, S. Derya Uysal, Andrew Ryan, and John L. Adams. 2016. Estimating causal effects for multivalued treatments: a comparison of approaches. *Statistics in Medicine* 35, 4 (2016), 534–552.

[12] Michael J. Lopez and Roee Gutman. 2017. Estimation of Causal Effects with Multiple Treatments: A Review and New Ideas. *Statist. Sci.* 32, 3 (Aug 2017), 432–454. https://doi.org/10.1214/17-sts612

[13] Romain Lopez, Chenchen Li, Xiang Yan, Junwu Xiong, Michael I. Jordan, Yuan Qi, and Le Song. 2019. Cost-Effective Incentive Allocation via Structured Counterfactual Inference. arXiv:1902.02495 [stat.ML]

[14] Daniel F. McCaffrey, Beth Ann Griffin, Daniel Almirall, Mary Ellen Slaughter, Rajeev Ramchand, and Lane F. Burgette. 2013. A tutorial on propensity score estimation for multiple treatments using generalized boosted models. *Statistics in Medicine* 32, 19 (2013), 3388–3414.

[15] Judea Pearl. 2009. *Causality: Models, Reasoning and Inference* (2nd ed.). Cambridge University Press, USA.

[16] Paul R. Rosenbaum and Donald B. Rubin. 1983. The Central Role of the Propensity Score in Observational Studies for Causal Effects. *Biometrika* 70 (1983), 41–55.

[17] Shiv Kumar Saini, Sunny Dhamnani, Aakash, Akil Arif Ibrahim, and Prithviraj Chavan. 2019. Multiple Treatment Effect Estimation Using Deep Generative Model with Task Embedding. In *The World Wide Web Conference* (San Francisco, CA, USA) *(WWW '19)*. Association for Computing Machinery, New York, NY, USA, 1601–1611.

[18] Patrick Schwab, Lorenz Linhardt, Stefan Bauer, Joachim M. Buhmann, and Walter Karlen. 2020. Learning Counterfactual Representations for Estimating Individual Dose-Response Curves. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI Press, 5612–5619.

[19] Anthony D. Scotina and Roee Gutman. 2019. Matching algorithms for causal inference with multiple treatments. *Statistics in Medicine* 38, 17 (2019), 3139–3167.

[20] Uri Shalit, Fredrik D. Johansson, and David Sontag. 2017. Estimating individual treatment effect: generalization bounds and algorithms. In *Proceedings of the 34th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, International Convention Centre, Sydney, Australia, 3076–3085.

[21] B. Sriperumbudur, K. Fukumizu, A. Gretton, B. Schölkopf, and G. Lanckriet. 2012. On the Empirical Estimation of Integral Probability Metrics. *Electronic Journal of Statistics* 6 (2012), 1550–1599.

[22] Bharath K. Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Bernhard Schölkopf, and Gert R. G. Lanckriet. 2009. On integral probability metrics, ϕ-divergences and binary classification. arXiv:0901.2698 [cs.IT]

[23] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. 2018. Representation Learning for Treatment Effect Estimation from Observational Data. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc., 2633–2643.

[24] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. 2018. GANITE: Estimation of Individualized Treatment Effects using Generative Adversarial Nets. In *International Conference on Learning Representations*.