Multi-treatment Effect Estimation from Biomedical Data

Raquel Aoki†, Yizhou Chen and Martin Ester
School of Computing Science, Simon Fraser University,
Vancouver, British Columbia, Canada
†*E-mail: raoki@sfu.ca*

Several biomedical applications contain multiple treatments from which we want to estimate the causal effect on a given outcome. Most existing Causal Inference methods, however, focus on single treatments. In this work, we propose a neural network that adopts a multi-task learning approach to estimate the effect of multiple treatments. We validated M3E2 in three synthetic benchmark datasets that mimic biomedical datasets. Our analysis showed that our method makes more accurate estimations than existing baselines.

Keywords: Causal Inference, Multiple-treatments, biomedical data

1. Introduction

Consider the following setting: an exploratory study on hearing loss as an Adverse Drug Reaction (ADR) in children under cancer treatment with the drug Cisplatin.[1] While Cisplatin is one of the most effective chemotherapeutic agents for children, reports have also demonstrated that 75-100% of infant patients have hearing loss. Note that patients often receive a drug cocktail, and while a single drug might not lead to ADR, ADR is observed when we have a combination of these drugs. Previous studies[1] pointed out that hearing loss is the result of a combination of factors, such as the patient's age, genetic predisposition, dosage, and exposure to several drugs (more drugs, more heavy metals accumulation in the body, higher the chances of hearing loss). The study's data are the patient's clinical information (low-dimensional), genetic information (high-dimensional), the drugs given to the patient, and the observed ADR.

In Causal Inference notation, the covariates X are the patients' clinical information and genetic information; the outcome of interest Y is the ADR, and each drug is a binary treatment ($\mathcal{T} = [T_0, T_1, ..., T_K]$, where $T_k = 1$ records that the k-th drug was given). Understanding and learning the causal effect of each treatment on the outcome can be used to support doctors in recommending more precise treatments, minimizing ADRs in this example, or maximizing the drug response in other cases. Note that existing treatment effect estimators designed for individual binary treatments could be adopted: For each drug $k \in \{0, ..., K\}$, we fit an estimator using all the other drugs as covariates. However, such an approach assumes the estimator would perform covariate adjustment correctly - and here is where we argue that an estimator that considers the multiple treatments together could be a better alternative for biomedical data.

Recent advances in Machine Learning(ML) are now widely being used to improve Causal Inference methodologies. One example is how ML can improve the covariate adjustment of applications with high-dimensional datasets. Such improvements fit perfectly with the precision medicine vision of developing diagnosis, prognosis, and treatment techniques that consider the individual, often high-dimensional data. Most machine learning methods solve only a single task, i.e. they predict a single target variable. Multi-task learning (MTL) methods,[2] on the other hand, optimize a model to simultaneously solve multiple tasks (or, in our context, treatments). The main argument in favor of MTL is that single-task learning may fail to capture the synergy of multiple treatments, e.g., an additive effect or a genetic predisposition to a certain combination of treatments, but not to individual treatment. Currently, there are only a few methods capable of estimating the causal effect of multiple treatments. Hi-CI[3] considers and models multiple treatments but assumes that only one is assigned to a unit at any given time. The Deconfounder Algorithm (DA),[4] a probabilistic graphical model, works with multiple treatments but has received some recent criticism regarding its assumptions.[5]

Contributions: The main contributions of this paper are as follows:

- We propose the Multi-gate Mixture-of-experts for Multi-treatment Effect Estimation (M3E2), a method to estimate the multi-treatment effect.
- We validate M3E2 in three synthetic datasets that mimic biomedical applications. We also compare our method with three existing baselines.
- We create the repository `github.com/raquelaoki/M3E2` with an implementation of our methods, baselines, and datasets. We also share all the configuration files for reproducibility of our results, with hyperparameters and seeds adopted.

2. Related Work

This work combines the estimation of treatment effects and multi-task learning (MTL).

Estimating Treatment Effects: BART,[6] Causal Forests,[7] CEVAE,[8] and Dragonnet,[9] have explored the estimation of a single treatment effect, using Bayesian Random Forests, Random Forests, VAEs, and neural networks (NN) respectively. The inverse propensity weighting-based methods,[10] meta-learners[11] also focused on binary single-treatments. The Deconfounder Algorithm,[4] Hi-CI,[3] approaches based on the propensity score,[12,13] and others[14–16] aim to estimate multi-treatment effect. However, many of these methods assume that only one treatment is applied to any given unit or consider all the combinatorial interventions, which is infeasible for larger numbers of treatments. Note that several works assume robustness to missing confounders.[4,8,14,17] Their robustness is often built on the assumption that extra information is known, such as a known number of hidden confounders or replacing unobserved confounders with proxies. There are, however, several concerns regarding some of these methods.[5,18] Our proposed method focuses on multiple treatment effect estimation through an outcome model in a multi-task learning neural network architecture and ignorability. By considering all treatments simultaneously, our proposed architecture can learn a better representation of input data and perform a better covariate adjustment than existing baselines.

Multi-task learning (MTL): MTL neural network (NN) architectures aim to optimize a single model for two or more tasks simultaneously. Hard-parameter sharing NN[19] is one

of the MTL pillars. Such architecture is composed of a set of layers shared among all tasks and a set of task-specific layers on the top. From the MTL perspective, the Dragonnet[9] has a hard-parameter sharing architecture. Building upon the hard-parameter sharing architectures, the Multi-gate Mixture-of-Experts (MMoE)[20] architecture, where each expert can be seen as a hard-parameter sharing NN, and all the experts are combined through a gate function, which is also trainable. The core idea of such an approach is to improve the model's generalization; plus, it allows experts to specialize in one of the tasks. To put into perspective, an MMoE is to hard-parameter sharing NN what a Random Forest Model is to a Decision Tree. Our proposed method M3E2 uses a MMoE[20] as a component. Our work expands the MMoE architecture to satisfy causal inference assumptions and estimate the multi-treatment effect.

3. MMoE for Multi-treatment Effect Estimation

This section describes our proposed method, M3E2. Its multi-task learning architecture simultaneously predicts the outcome and the propensity scores for each treatment.

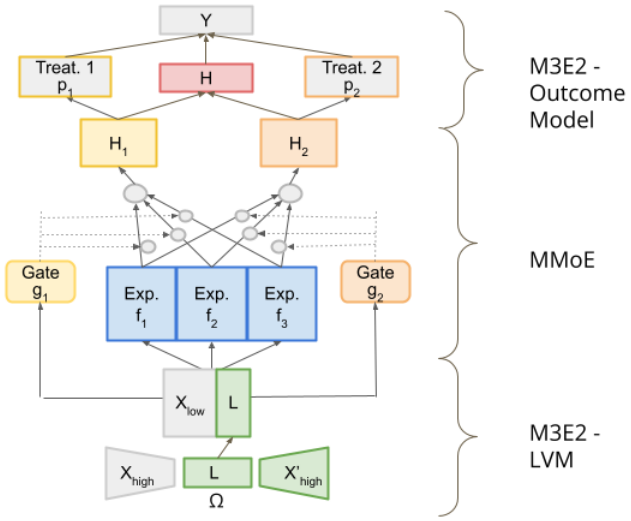

Fig. 1. M3E2 training architecture, for $K = 2$ (two treatments), and 3 experts. It receives as input the covariates $X = [X_{low}, X_{high}]$, and predicts the treatment assignment $\mathcal{T} = \{T_1, ..., T_K\}$ and the outcome Y. The LVM model Ω learns a latent representation L of the high-dimensional covariates X_{high}. The gates g_k, experts $f_e, \forall e \in \{1, 2, 3\}$, and task-specific layers H_1 and H_2 learn a representation H of the input data, and H is used to predict the propensity scores p_1 and p_2 and the outcome Y.

When working with observational studies, one must always describe how the confounders are addressed. Some works assume no unobserved confounders,[6,9,21,22] others try to reduce the bias through latent variables;[4,8,17] while others question if the latent variables are solving the problem at all.[5,18] While exploring alternatives to the ignorability assumption is an interesting research direction, the main focus of this work is the estimation of effect of multiple treatments. Hence, in our work, we assume no unobserved confounders.

Figure 1 illustrates the proposed neural network architecture, with a MMoE,[20] and a Latent Variable Model (LVM) as subcomponents. This architecture predicts $K + 1$ tasks: the

outcome Y and K propensity scores p_k. The propensity scores estimate the probability of a treatment being assigned given the covariates ($P(T_k = 1|X)$), and it is important to guarantee the identifiability of the causal effects (Theorem 1). The LVM contributes to the model by efficiently combining low and high-dimensional covariates (section 3.2). The MMoE is an MTL architecture adopted to handle multiple tasks. It contains a combination of experts, gates, and task-specific layers (section 3.3).

One of the strengths of M3E2 is its capacity to estimate the combined effect of a large number of treatments: the M3E2 network only grows linearly with the number of treatments, handling all potential combinations, something that other multi-treatment methods typically struggle to accomplish. Furthermore, the proposed architecture of M3E2 extends the MMoE architecture by incorporating causal inference assumptions through suitable regularizers and adding the outcome model to estimate the treatment effects.

Notation: We define low-dimensional covariates as X_{low} and high-dimensional covariates as X_{high}. An example of the first is clinical variables and, from the latter, genomics information. The split of covariates into low-dimensional and high-dimensional will be explained in Section 3.2. We define the covariates concatenation as $X = [X_{low}, X_{high}]$. The continuous outcome is Y, and K represents the number of treatments. $\mathcal{T} = \{T_0 = t_0, T_1 = t_1, ..., T_K = t_K\}$, where $\mathcal{T}$ could e.g. be the *drug cocktail* taken by a patient.

3.1. *Assumptions*

Assumption 1. Stable Unit Treatment Value Assumption (SUTVA):[23] the response of a particular unit depends only on the treatment(s) assigned, not the treatments of other units.

Assumption 2. Common Confounders and conditional independence:[24] Treatments share confounders. Given the shared confounders, the treatments are independent.

$$T_i \perp T_j | X, \forall i, j \in \{0, ..., K\}, i \neq j$$

Assumption 3. Ignorability - the potential outcome is independent of the treatments given the covariates.

Theorem 1. *Sufficiency of Propensity Score:[9,25] If the average treatment effect is identifiable from observational data by adjusting for X, i.e., $ATE = \mathbb{E}_X[\mathbb{E}_Y[Y|X, T = 1] - \mathbb{E}_Y[Y|X, T = 0]]$, then adjusting for the propensity score also suffices:*

$ATE = \mathbb{E}_X[\mathbb{E}_Y[Y|h(X), T = 1] - \mathbb{E}_Y[Y|h(X), T = 0]]$

First, we consider applications with a continuous outcome, binary or continuous treatments, and a set of covariates. Assumption 1 (SUTVA) is standard in Causal Inference. According to SUTVA, the samples are independent and do not interfere with each other. Assumptions 2 and 3 are related to the identifiability of the treatment effect. Assumption 2 assumes no links (dependencies) between the treatments given the covariates, and Assumption 3 assures all back-door paths can be blocked by conditioning on the observed covariates X - guaranteeing the identifiability of the treatment effect.[26] Assumption 2 is also related to multi-task learning (MTL). The ideal use of MTL is when tasks (in our case, treatments) are somehow related. In

that case, it is reasonable to assume they also share confounders. The Theorem 1 is presented here as originally proposed, so for the proofs and demonstrations, please check the original publications.[25,27] According to Theorem 1, it suffices to adjust only the information in X that is relevant for predicting the treatment T_k, which is the output of $H_k(X_{L1})$. For multiple treatments, the generalization goes as follows:[27]

$ATE = E[E[Y|H(X_{L1}), T_1 = t_1, ..., T_K = t_K] - E[Y|H(X_{L1}), T_1 = 1 - t_1, ..., T_K = t_K]]$

Under these assumptions and theorem, the identifiability comes from the Propensity Score's Sufficiency and the following causal structure: $\mathcal{T} \to Y$, $X \to \mathcal{T}$, $X \to Y$.

3.2. *Latent Variable Model (LVM)*

M3E2 can handle different data types by dividing the input covariates X into two groups, X_{low} and X_{high}. While the Latent Variable Model (LVM) handles the covariates in X_{high}, the X_{low} covariates are fed directly to the experts. The split of the covariates X into X_{low} and X_{high} is defined by the user. Ideally, X_{high} contains high-dimensional covariates, such as gene expression, single-cell data, or image data; and X_{low} contains low-dimensional data, such as clinical variables. Note that, in applications with only one data type, both $X_{low} = \emptyset$ and $X_{high} = X$, and $X_{low} = X$ and $X_{high} = \emptyset$ are acceptable splits.

In applications where $X_{high} \neq \emptyset$, M3E2 uses a LVM to reduce the dimensionality of the covariates in X_{high}. Note that, while there are similarities with other works that adopt proxies to handle unobserved confounders, our LVM component is responsible only for reducing the dimensionality of X_{high}. As described in Section 3.1, our work assumes strong ignorability, a setting with no unobserved confounders. Under strong ignorability, however, we can still have confounding within the observed data. The LVM component, along with the experts, is responsible for extracting a meaningful representation of the input data. These features are used in the covariate adjustment $E[Y|X, T_0, ..., T_k]$, which should close the back-doors and make the treatment effect identifiable. To learn a meaningful representation of X in applications with a mix of high-dimensional and low-dimensional covariates, it was important to find an approach that is capable of combining these different types of covariates. Without the LVM component, the experts could give a disproportional weight to X_{high} covariates, as they would be the majority in X, and even ignore relevant information in X_{low}.

In our experiments, M3E2 adopts an autoencoder with two linear encoder layers and two linear decoder layers. Note, however, that one is free to choose a different architecture or factor model to extract a latent representation of X_{high}. Consider an application with n samples, c_2 columns in X_{high}, c_L as the latent variables size, and the input data X_{high} as a matrix $n \times c_2$. The function $\omega_{enc}(X_{high})$ returns $L_{(n \times c_L)}$, a representation of X_{high} in a lower dimension. Finally, $\omega_{dec}(X_{high})$ returns the reconstructed data X'_{high}, back on $n \times c_2$ space.

3.3. *MMoE Architecture*

In Machine Learning, it is common for a set of shared layers to predict multiple tasks. These architectures are called hard-parameter sharing neural networks. A multi-gate mixture-of-expert (MMoE)[20] architecture contains several experts, where each expert can be seen as a hard-parameter sharing neural network. It was shown that MMoE architectures generalize

better,[20] especially in biological applications.[28]

The user defines the number of experts E and the f_e architecture. In the context of multiple treatment effect estimation, the tasks are the propensity score and the outcome Y prediction. The experts' input data is $X_{L1} = [\Omega_{enc}(X_{high}), X_{low}] = [L, X_{low}]$. The ideal number of experts depends on the tasks. Homogeneous tasks might not benefit from many experts and might overfit if the number of experts is too large. Conversely, heterogeneous tasks tend to benefit from a larger number of experts. Note that the definition of homogeneous and heterogeneous tasks is subjective. Here, we define applications whose tasks adopt the same loss as homogeneous tasks. An example would be an application with only classification tasks. On the other hand, heterogeneous task applications contain classification, regression, multi-label, and other potential tasks in the MTL model. The gates control the contribution of each expert to each task. There is a gate g_k per treatment defined as: $g_k(X_{L1}) = softmax(W_K \times X_{L1}), \forall k \in 1, ..., K$, where $W_k \in R^{E \times d}$ is a trainable matrix of weights, E is the number of experts defined by the user, and d is the number of columns in X_{L1}. Finally, note that the gates can be seen as an attention[29] mechanism, learning which experts are more relevant for each task.

3.4. *Task-specific Layers*

The task-specific layers are responsible for predicting the propensity score p_k and the outcome of interest $\hat{Y}$. Each treatment task-specific layer receives as input a weighted average of the experts, where the weights come from the gates associated with that given task. This relationship is formally defined as:

$H_k = h_k(\sum_{e=1}^{E} g_k(X_{L1}) f_e(X_{L1})), \forall k \in \{1, ..., K\}$

In the training phase (Figure 1), the treatment assignment is predicted with the propensity score p_k, estimated as $p_k = P(T_k = t|H_k)$ (for discrete treatments) or $p_k = P(T_k \leq t|H)$ (for continuous treatments using the conditional density $f_{T|X}(t, x)$[30,31]). To estimate the treatment assignment of T_k we only use $H_k, \forall k \in \{1, ..., K\}$. For binary treatments, a softmax activation function will outputs, for each sample, the probability of $P(T_k = 1|H_k)$ and $P(T_k = 0|H_k)$. These predictions are used to calculate the loss of the neural network, as described in Section 3.5. The propensity score losses are used to drive H_k to be sufficient (Theorem 1 - Section 3.1). Note that h_k can be a combination of one or more layers.

Finally, a layer with trainable weights Φ is used to predict the outcome. Consider the input data of this layer as $X_{TH} = [T_1, ..., T_K, H]$, where $T_1, ..., T_K$ are the observed treatment assignments, $H = \frac{\sum_{k=1}^{K} H_k}{K}$, and c_{TH} is the number of columns. The trainable weights layer $\Phi = [\tau_1, \ldots, \tau_k, \ldots, \tau_{c_{TH}}]$ estimates the final outcome as $Y = \Phi \times X_{TH}$. In our context of treatment effect estimation, τ_k is the treatment effect of the treatment k. The Φ works as an outcome model and each weight associated with a $T_i, \forall i \in \{0, ..., K\}$ represents an ATE_i.

Our approach targets additive effect, which is fairly common in biomedical applications.[32] Consider, for example, the ADR study on patients under cancer therapy described in Section 1. Many of these drugs contain heavy metals, and their accumulation can result in adverse drug reactions. Non-linear effects[a] are an interesting extension left for future work.

[a]Note that the linearity only applies to the last layer Φ, not to the autoencoder or the experts.

3.5. *Loss function*

M3E2's loss function is composed of:

(1) Root mean square error loss $\ell_y(Y, \hat{Y}) = RMSE(Y, \hat{Y})$ for continuous outcomes and binary cross-entropy $\ell_y(Y, \hat{Y}) = BCE(Y, \hat{Y})$ for binary outcomes.
(2) Similar to the outcome loss functions, we adopt $\ell_{p_k}(T, T') = RMSE(T_k, \hat{T}_k)$ or/and $\ell_{p_k}(T, T') = BCE(T_k, \hat{T}_k)$ as the propensity score losses, $\forall k \in \{0, ..., K\}$.
(3) $\ell_A(X_{high}, X'_{high}) = RMSE(X_{high}, X'_{high})$ is the autoencoder loss function.
(4) $\frac{1}{2n} \sum_w w^2$ as the L_2 regularization.

As a reminder, while our architecture minimizes the propensity score and the outcome losses, our main target is to obtain estimates of the treatment effects. The treatment effects are a co-product of this model, i.e., the weights associated with the treatments in the trainable layer Φ (See Section 3.4). The model also learns weights in Φ associated with the H; however, these are not considered treatment effects. The total loss is $\mathcal{L} = \alpha\ell_y + \beta \sum_k^K \ell_{p_k} + \gamma\ell_A + \frac{\lambda}{2n} \sum_w w^2$, where α, β and γ are weights. There are two possible ways to define these weights: to adopt them as a hyper-parameter or to adopt an MTL task balancing approach. Modifying both ℓ_{g_k} and ℓ_y to other loss functions is also straightforward.

4. Experiments

In causal inference, the lack of ground truth for real-world applications poses a challenge to its evaluation. Therefore, we adopt three synthetic datasets that have known treatment effects. These synthetic datasets mimic existing biomedical datasets:

- Genome-Wide Association Study (GWAS):[4,33,34] Semi-synthetic sparse dataset with 1000 covariates, 3-10 binary treatments, and continuous outcome. In this dataset, the covariates and treatments are single-nucleotide polymorphisms (SNPs), and the outcome represents a clinical trait. The simulation starts by removing highly correlated SNPs with linkage disequilibrium from the 1000 Genome Project (TGP).[35] Then, a PCA extracts $c = 5$ components from TGP, creating the genetic representation matrix $\Gamma_{v,c}$. The patients' representation matrix is generated as $\Pi_{n,c} \sim 0.9 \times Uniform(0, 0.5)$, where n is the number of desire samples. The covariates are simulated as $X_{n,v} \sim Binomial(1, \Pi_{n,c} \times \Gamma^T_{v,c})$. The set $\mathcal{K}$ contains the index of K columns randomly picked to be treatments. The effect of each covariate is defined as $\tau_i \sim Normal(0, 0.5) \forall i \in \mathcal{K}$(causal effect), else, $\tau_i = 0$ (non-causal effect). Three groups were extracted using k-means(X) to add confounding. Each group $l \in \{1, 2, 3\}$ has an intercept value λ_l and noise distribution $\epsilon \sim Normal(0, \sigma_l)$, $\sigma_l \sim InvGamma(3, 1)$. The outcome is calculated as $Y = \sum_v \tau_v X_{n,v} + \lambda_{l_n} + \epsilon$.
- Copula:[32] This recently proposed dataset also mimics a Genome-Wide Association Study. The Copula, unlike the GWAS dataset, features a fully synthetic dataset. We adopted the setting with four treatments and non-linear outcomes. The covariates are generated as $X_{n,v} \sim Normal(0, \sigma)$, where n is the sample size and v the number of covariates. The treatments are simulated as $T_{n,l} = PCA_1(X_{n,v}) + \epsilon_t, \forall l \in \{1, 2, 3, 4\}$, $\epsilon_t \sim Normal(0, \sigma_t)$, and $Y = 3 \times T_1 - T_2 + T_3 I_{T_3>0} + 0.7 \times T_3 I_{T_3\leq 0} - 0.06 \times T_4 - 4 \times T_1^2 + 2.8 \times \sum_v X_{n,v} + \epsilon_y$, $\epsilon_y \sim Normal(0, \sigma_y)$. The causal effects are $\tau = [1, 0.25, -0.2, 0.1]$.

- IHDP:[6,8,9] the Infant Health and Development Program (IHDP) is a traditional benchmark for single binary treatments. It is supposed to mimic a study on infant development. In that study, the treatment was assigned ($T = 1$) if the child had special care/home visits from a trained provider. The outcome Y is cognitive test scores, and the goal is to measure the causal effect of the home visits. This benchmark contains ten replications of such a study, with 24 covariates and a continuous outcome. We adopt this dataset to compare our proposed method with some of the single-treatment baselines that have been previously evaluated on the IHDP benchmark datasets. [b]

Due to the synthetic nature of the datasets adopted [c], we can calculate the mean absolute error (MAE) between the estimated treatment effect and the true treatment effect. Defining τ_k as the true treatment effect of T_k, and $\hat{\tau}_k$ as its estimated value by one of the methods. As we have multiple treatment effects, we report their average error $\frac{\sum_{k=0}^{K} |\tau_k - \hat{\tau}_k|}{K}$, where K is the total number of treatments. We repeat each combination of ($data \times model \times setting$) $B = 20$ times, and in our plots, we show the MAE calculated over all these runs:

$$MAE = \sum_{b=0}^{B} \left(\frac{\sum_{k=0}^{K} |\tau_k - \hat{\tau}_k|}{K} \right) \frac{1}{B} \tag{1}$$

A good estimator has estimates close to the true treatment effect values; therefore, *low MAE values are desirable*. We adopt an experimental setting similar to the multi-task learning settings,[20] where the proposed multi-task learning method is compared with other multi-task learning methods and single-task learning models. Among our baselines, the DA[4] is the only method that can estimate the effect of multiple treatments with one model. The CEVAE[8] and Dragonnet[9] are single-treatment methods. We used the author's implementation of the baselines when available. For single-treatment baselines, the multiple treatment effects were estimated as follows: to estimate τ_1, the baseline methods receive as input T_1 as the treatment assignment, and the columns $T_0, T_2, ..., T_K$ are added to X_{low}. We follow this setup for all K treatments. We also performed experiments with BART. However, since CEVAE and Dragonnet achieved better performance results in the recent publications,[8,9] and BART performed poorly on the GWAS and Copula datasets, we decided not to discuss BART in the experimental section.

4.1. *Overall Performance*

Figure 2 shows, for each dataset, the average MAE across all settings. Our proposed method, M3E2, clearly outperforms all baselines on the multi-treatment datasets GWAS and COPULA. On IHDP, a single-treatment dataset, M3E2 was outperformed by Dragonnet, yet, it was better than the other two baselines. Note that our results for Dragonnet on IHDP match the results previously reported,[9] and the estimators' larger variance on the IHDP dataset can be explained by the scale of the true treatment effect. Our main take from Figure 2 is that our method outperforms all the baselines on its ideal use-case: applications with multiple treatment effects.

[b]Implementation available at `github.com/AMLab-Amsterdam/`
[c]Implementation available at `github.com/raquelaoki/CompBioAndSimulated_Datasets`

In single-treatment applications, while achieving reasonable results, simpler architectures that target single-treatment estimation like the Dragonnet tend to achieve better performance.

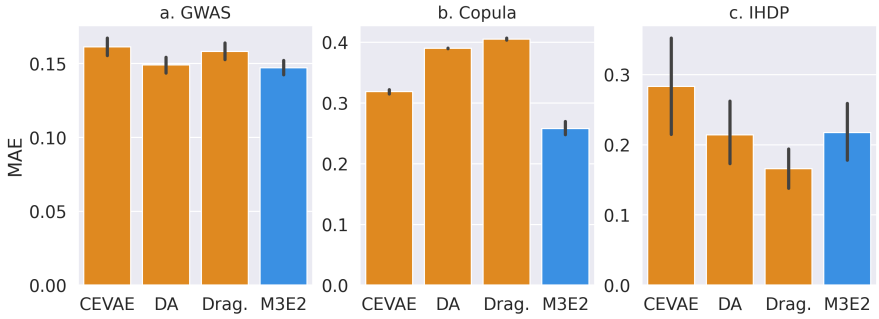

Fig. 2. MAE barplots of the M3E2 and baseline methods. Small MAE values are desirable. The black line indicates a 95% confidence interval.

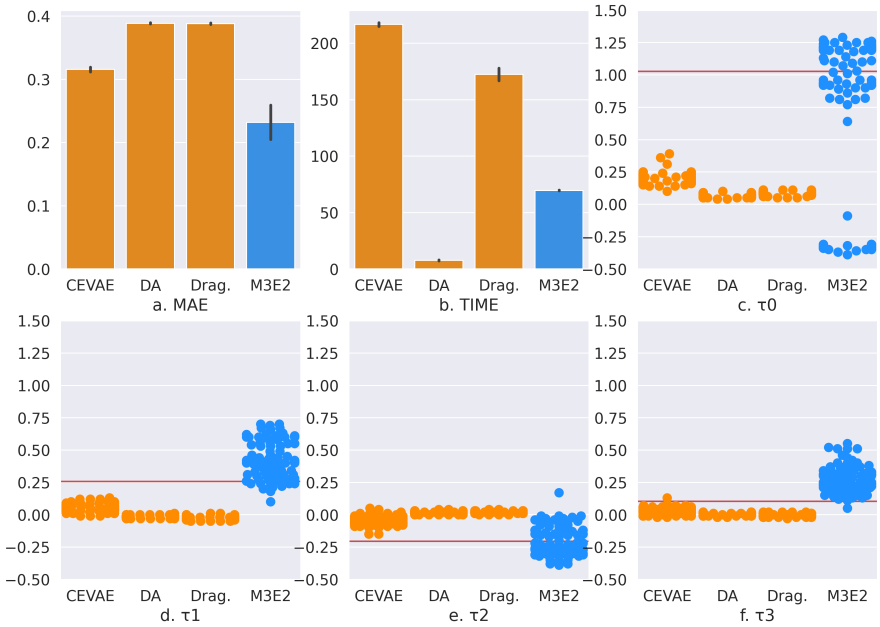

Fig. 3. Copula results for one simulated dataset ($n = 10000, k = 4, v = 10$) with 24 independent repetitions of each model. The baselines' results are shown in orange, our results are in blue, and the red line shows the true effect (c-f).

Figure 3 shows a deeper analysis of the Copula dataset. Figure 3.a shows that M3E2 has the lowest MAE values compared to the other baselines. Figure 3.b shows the total run time of each method in seconds. As a reminder, both DA and M3E2 fit one model for all treatments; Dragonnet and CEVAE, on the other hand, fit one model for each treatment. DA, a probabilistic model, has the fastest running time; M3E2 has the lowest running time among the NN methods. A comparison between the true τ (line in red) and the estimated treatment effects (dots) is shown in Figures 3.c-f. Note that for τ_0 and τ_2, M3E2 is the only method

whose estimates are centered around the true value. For τ_1 and τ_3, M3E2 overestimates the treatment effects, yet, it still produces reasonably good estimates. Overall, M3E2 has a good performance. However, we noticed two limitations: First, M3E2 has a larger variance than the other methods; second, for some runs, it estimated values very far from the true treatment effect τ_0. Considering our baselines, while they have a smaller variance, we noticed that DA and Dragonnet often estimated the treatment effect as 0, indicating that these methods might fail to estimate the treatment effect in this dataset correctly, despite achieving reasonable predictive performance. CEVAE was the second-best method; still, its results were never centered around the true values (red lines) and often underestimated the magnitude of the treatment effect.

4.2. *Impact of Dataset Parameters*

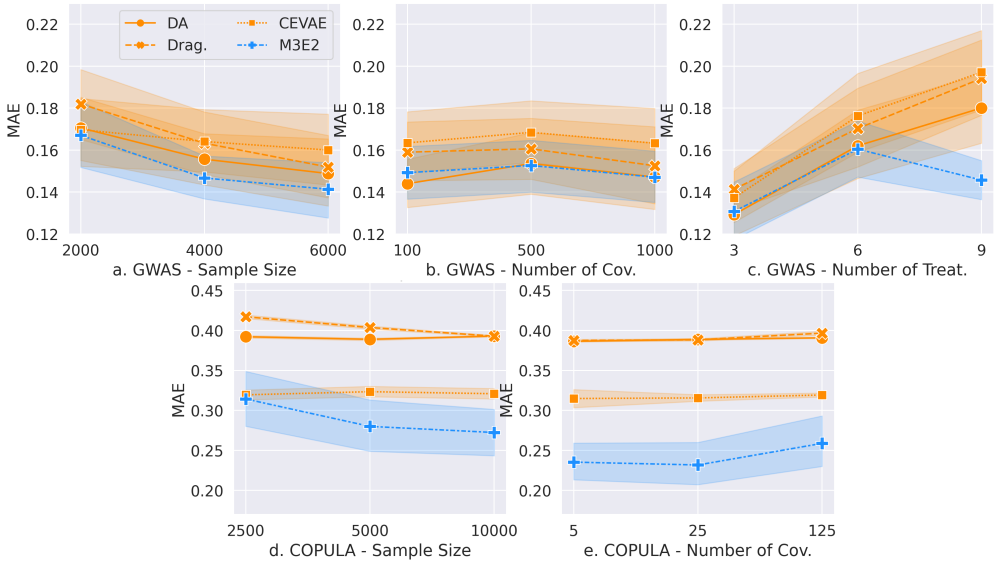

Fig. 4. Impact of the dataset parameters in estimating multiple treatment effects.

We also explored the impact of the dataset parameters in estimating the multiple treatment effects. We focused on three parameters: the sample size, number of treatments, and covariates. Figure 4 shows, in detail, the average MAE and the 95% confidence interval (colored area) for the several settings. Figure 4.*a* and 4.*d* show the impact of the sample size on the GWAS and Copula dataset, respectively. Our proposed method, M3E2, is the method that benefits the most from increasing the sample size. We noticed that all methods are robust to the increase in the number of covariates (Figures 4.b and 4.e), with M3E2 having a small increase on MAE on the Copula dataset with 125 covariates. The most surprising result of all is shown in Figure 4.c. The MAE increases in all baselines with the increase in the number of treatments. Nevertheless, M3E2 achieves better results with nine treatments than with six treatments. Such a result shows that, while the methods are similar regarding the dataset impact on MAE and are quite robust to variations in the number of covariates, M3E2 significantly outperforms all other methods when a larger number of treatment effects are considered.

5. Discussion and Conclusion

In this paper, we have investigated the problem of estimating the effect of multiple treatments in observational data, a setting often found in biomedical applications. To address current limitations, we proposed the M3E2, a multiple treatment effect estimator that uses a MTL neural network architecture. One of the main advantages of M3E2 is its flexibility, as several of its subcomponents can be replaced by alternative implementations, e.g., by different experts, latent variable models, or propensity score predictors. We experimentally compared M3E2 against three baselines on three synthetic benchmark datasets that mimic biomedical applications. The online repository `github.com/raquelaoki/M3E2` contains the code to replicate all the experiments, and we put extra effort into making the M3E2 implementation agnostic to the application; therefore, its deployment in other applications should be straightforward. M3E2 demonstrated promising experimental results and strong evidence that MTL contributed to more accurate estimates of the treatment effects. Nevertheless, there remain several directions for future research. As discussed in Section 3.1, our method assumes ignorability, which is quite limiting in real-life applications. M3E2 also inherits the limitations of other MTL models, in particular, the susceptibility to imbalanced tasks and overfitting. All strengths and limitations considered, we believe that M3E2 has a very good use case with manageable limitations. In future research, we want to apply our proposed method to a real-world dataset that records adverse drug reactions in therapies for treating cancer in infants, moving a step forward toward the precision medicine goal of providing the *right drug at the right dose to the right patient*.[36]

References

1. B. I. Drögemöller, G. E. Wright, C. Lo, T. Le, B. Brooks, A. P. Bhavsar, S. R. Rassekh, C. J. Ross and B. C. Carleton, Pharmacogenomics of cisplatin-induced ototoxicity: Successes, shortcomings, and future avenues of research, *Clinical Pharmacology & Therapeutics* **106**, 350 (2019).
2. S. Ruder, An overview of multi-task learning in deep neural networks, *arXiv preprint arXiv:1706.05098* (2017).
3. A. Sharma, G. Gupta, R. Prasad, A. Chatterjee, L. Vig and G. Shroff, Hi-ci: Deep causal inference in high dimensions, in *Proceedings of the 2020 KDD Workshop on Causal Discovery*, 2020.
4. Y. Wang and D. M. Blei, The blessings of multiple causes, *Journal of the American Statistical Association* , 1 (2019).
5. A. D'Amour, On multi-cause causal inference with unobserved confounding: Counterexamples, impossibility, and alternatives, *arXiv preprint arXiv:1902.10286* (2019).
6. J. L. Hill, Bayesian nonparametric modeling for causal inference, *Journal of Computational and Graphical Statistics* **20**, 217 (2011).
7. S. Wager and S. Athey, Estimation and inference of heterogeneous treatment effects using random forests, *Journal of the American Statistical Association* **113**, 1228 (2018).
8. C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel and M. Welling, Causal effect inference with deep latent-variable models, in *NeurIPS*, 2017.
9. C. Shi, D. Blei and V. Veitch, Adapting neural networks for the estimation of treatment effects, in *NeurIPS*, 2019.
10. M. A. Hernán and J. M. Robins, Estimating causal effects from epidemiological data, *Journal of Epidemiology & Community Health* **60**, 578 (2006).
11. A. Curth and M. van der Schaar, Nonparametric estimation of heterogeneous treatment effects: From theory to learning algorithms, in *AISTATS*, 2021.

12. M. Lechner, Identification and estimation of causal effects of multiple treatments under the conditional independence assumption, 43 (2001).
13. M. J. Lopez and R. Gutman, Estimation of causal effects with multiple treatments: a review and new ideas, *Statistical Science* , 432 (2017).
14. W. Miao, W. Hu, E. L. Ogburn and X. Zhou, Identifying effects of multiple treatments in the presence of unmeasured confounding, *Journal of the American Statistical Association* , 1 (2021).
15. A. Tanimoto, T. Sakai, T. Takenouchi and H. Kashima, Regret minimization for causal inference on large treatment space, in *AISTATS*, 2021.
16. Z. Qian, A. Curth and M. van der Schaar, Estimating multi-cause treatment effects via single-cause perturbation, in *NeurIPS*, 2021.
17. A. Mastouri, Y. Zhu, L. Gultchin, A. Korba, R. Silva, M. J. Kusner, A. Gretton and K. Muandet, Proximal causal learning with kernels: Two-stage estimation and moment restriction, *NeurIPS* (2021).
18. S. Rissanen and P. Marttinen, A critical look at the consistency of causal estimation with deep latent variable models, *NeurIPS* **34** (2021).
19. R. Caruana, Multitask learning: A knowledge-based source of inductive bias, *ICML* (1993).
20. J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong and E. H. Chi, Modeling task relationships in multi-task learning with multi-gate mixture-of-experts, in *ACM SIGKDD*, 2018.
21. U. Shalit, F. D. Johansson and D. Sontag, Estimating individual treatment effect: generalization bounds and algorithms, in *ICML*, 2017.
22. A. N. Glynn and K. M. Quinn, An introduction to the augmented inverse propensity weighted estimator, *Political analysis* **18**, 36 (2010).
23. D. B. Rubin, Randomization analysis of experimental data: The fisher randomization test comment, *Journal of the American Statistical Association* **75**, 591 (1980).
24. R. Ranganath and A. Perotte, Multiple causal inference with latent confounding, *arXiv preprint arXiv:1805.08273* (2018).
25. P. R. Rosenbaum and D. B. Rubin, The central role of the propensity score in observational studies for causal effects, *Biometrika* **70**, 41 (1983).
26. J. Pearl, Causal diagrams for empirical research, *Biometrika* **82**, 669 (1995).
27. G. W. Imbens, The role of the propensity score in estimating dose-response functions, *Biometrika* **87**, 706 (2000).
28. R. Aoki, F. Tung and G. L. Oliveira, Heterogeneous multi-task learning with expert diversity, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* (2022).
29. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, *NeurIPS* **30** (2017).
30. K. Hirano and G. W. Imbens, The propensity score with continuous treatments, *Applied Bayesian modeling and causal inference from incomplete-data perspectives* **226164**, 73 (2004).
31. L. Nie, M. Ye, qiang liu and D. Nicolae, Varying coefficient neural network with functional targeted regularization for estimating continuous treatment effects, in *ICLR*, 2021.
32. J. Zheng, A. D'Amour and A. Franks, Copula-based sensitivity analysis for multi-treatment causal inference with unobserved confounding, *arXiv preprint arXiv:2102.09412* (2021).
33. M. Song, W. Hao and J. D. Storey, Testing for genetic associations in arbitrarily structured populations, *Nature genetics* **47**, 550 (2015).
34. R. Aoki and M. Ester, Parkca: Causal inference with partially known causes, *Pac Symp Biocomputing* (2021).
35. G. P. Consortium, A. Auton, L. Brooks, R. Durbin, E. Garrison and H. Kang, A global reference for human genetic variation, *Nature* **526**, 68 (2015).
36. F. S. Collins and H. Varmus, A new initiative on precision medicine, *New England journal of medicine* **372**, 793 (2015).