Review of Deep Learning Methods for Individual Treatment Effect Estimation with Automatic Hyperparameter Optimization

Marcus Buchwald [1,1], Andrei Sirazitdinov [2], Jürgen Hesser [2], and Vincent Heuveline [2]

[1]Mannheim Institute for Intelligent Systems in Medicine (MIISM)
[2]Affiliation not available

October 31, 2023

Abstract

Abstract—Estimation of individual treatment effect (ITE) for different types of treatment is a common challenge in therapy assessments, clinical trials and diagnosis. Deep learning methods, namely representation based, adversarial, and variational, have shown promising potential in ITE estimation. However, it was unclear whether the hyperparameters of the originally proposed methods were well optimized for different benchmark datasets. To solve these problems, we created a public code library containing representation-based, adversarial, and variational methods written in TensorFlow. In order to have a broader collection of ITE estimation methods, we have also included neural network based meta-learners. The code library is made accessible for reproducibility and facilitating future works in the field of causal inference. Our results demonstrate that performance of most methods can be improved using automatic hyperparameter optimization. Additionally, we review the methods and compare the performance of the optimized models from our library on publicly available datasets. The potential of hyperparameter optimization may encourage researchers to focus on this aspect when creating new methods for inferring individual treatment effect.

Review of Deep Learning Methods for Individual Treatment Effect Estimation with Automatic Hyperparameter Optimization

Andrei Sirazitdinov, Marcus Buchwald, Jürgen Hesser, Vincent Heuveline, *Members, IEEE*

***Abstract*—Estimation of individual treatment effect (ITE) for different types of treatment is a common challenge in therapy assessments, clinical trials and diagnosis. Deep learning methods, namely representation based, adversarial, and variational, have shown promising potential in ITE estimation. However, it was unclear whether the hyperparameters of the originally proposed methods were well optimized for different benchmark datasets. To solve these problems, we created a public code library containing representation based, adversarial, and variational methods written in TensorFlow. In order to have a broader collection of ITE estimation methods, we have also included neural network based meta-learners. The code library is made accessible for reproducibility and facilitating future works in the field of causal inference. Our results demonstrate that performance of most methods can be improved using automatic hyperparameter optimization. Additionally, we review the methods and compare the performance of the optimized models from our library on publicly available datasets. The potential of hyperparameter optimization may encourage researchers to focus on this aspect when creating new methods for inferring individual treatment effect.**

***Index Terms*—Causal Inference, Deep Learning, Individual Treatment Effect (ITE) Estimation.**

I. Introduction

CAUSAL inference addresses the question of what would be the outcome if instead of one treatment, an alternative one was applied. In the general context of individual treatment effect estimation, this requires modifying the treatment prescription to measure differences in outcomes. Sources of data for causal inference include randomized control trials (RCT) and observational studies. An elementary problem in this case is the lack of the unseen or counterfactual treatment outcomes [1]. In RCT, a person is assigned to the treatment - or control group at random, which supports the assumption that the treatment assignment does not depend on the individual characteristics of the patient. In the case of an ideal RCT with a sufficiently large patient set, one might easily compute a conditional average treatment effect. It acts like an individual treatment effect, as the distributions of patients with different features in the treatment group and in the control group are sufficiently similar. In reality, RCT is limited through ethical and financial aspects. For example, it is unethical to assign surgical interventions to random participants. Observational studies on the other hand rely on existing patient data, i.e. clinical records, and are as such easier to conduct [2]. The main drawback of observational studies is that treatment assignment is highly correlated with subject characteristics. Especially if the treatment assignment algorithm is unknown, potential hidden factors impede the estimation of ITE [3]. In this case, computation of the treatment outcome is biased.

In consequence, to make a prediction of treatment outcome in observational studies possible, we rely on three assumptions [4]. The first assumption is called *unconfoundness*. It presumes the absence of hidden, i.e. non-available, variables influencing both treatment and outcome. The second assumption is *positivity*, which states that there must be a non-zero probability of receiving a treatment. This means that, a priori, one cannot infer whether a patient belongs to the treatment or control group based on one's covariates. As a consequence, the counterfactual outcomes from all subspaces of the covariate space can be inferred [5]. The third assumption is *consistency* meaning for patients with similar characteristics receiving the same treatment, the same outcome is expected.

The main problem of observational studies is group imbalance. Study cases often include significantly more control patients than treated. Another problem lies in a highly nonlinear dependency between covariates and outcome for most real-world datasets [6]. In this case we can not rely on classical methods of outcome prediction such as linear regression. Neural networks have performed exceptionally well in the case of nonlinear and linear relationships between input and outcome. They are used in many areas, such as working with text or images [7].

In this paper, we provide an overview of a variety of estab-

The authors gratefully acknowledge the data storage service SDS@hd supported by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK) and the German Research Foundation (DFG) through grant INST 35/1314-1 FUGG and INST 35/1503-1 FUGG. This work is funded by the German federal projects LeMeDaRT (01ZZ2105A) and PerPain (01EC1904B).

Andrei Sirazitdinov is affiliated with the Mannheim Institute for Intelligent Systems in Medicine, 68167, Mannheim, Germany (e-mail: andrei.sirazitdinov@medma.uni-heidelberg.de).

Marcus Buchwald is affiliated with the Mannheim Institute for Intelligent Systems in Medicine, Engineering Mathematics and Computing Lab (EMCL), Interdisciplinary Center for Scientific Computing (IWR), and Heidelberg Institute for Theoretical Studies (HITS) all located in 69120, Heidelberg, Germany, (e-mail: marcus.buchwald@medma.uni-heidelberg.de).

Jürgen Hesser is affiliated with the Mannheim Institute for Intelligent Systems in Medicine, 68167, Mannheim, Germany, Medical School, Heidelberg University, 69117, Heidelberg, Germany, and Interdisciplinary Center for Scientific Computing (IWR), Central Institute for Computer Engineering (ZITI), CZS Heidelberg Center for Model-Based AI all located in 69120, Heidelberg, Germany (e-mail: juergen.hesser@medma.uni-heidelberg.de).

Vincent Heuveline is affiliated with Engineering Mathematics and Computing Lab (EMCL), Interdisciplinary Center for Scientific Computing (IWR), Heidelberg Institute for Theoretical Studies (HITS) all located in 69120, Heidelberg, Germany (e-mail: vincent.heuveline@uni-heidelberg.de).

Andrei Sirazitdinov and Marcus Buchwald equally contributed to this work.

lished neural networks based methods used for ITE estimation. We review their architectures and evaluate them on publicly available datasets. Strong and weak points of each architecture are discussed. Finally, a summary of the results is provided. We also offer an extensive causal inference library and outline future directions for building ITE estimation methods based on neural networks.

II. Related Work

Recently, causal inference methods utilizing neural networks enjoy increasing popularity [8]–[14]. These can be divided into representation based, adversarial and variational methods. Representation based methods use neural networks to transform data into often lower-dimensional latent space. This facilitates inference as representations for treated and untreated subjects are located closer to each other compared to input space [8], [9]. Adversarial based strategies employ Generative Adversarial Networks (GAN) [15]. They learn to generate artificial treatment predictions while simultaneously being trained to discriminate between them and the true outcomes in a competing strategy to improve outcome estimation performance [10], [12]. Variational methods [11], [13], [14] use variational autoencoders [16] to convert the input into latent space and then using the latter to sample the treatment outcomes, including uncertainties. Such methods rely on Directed Acyclic Graphs (DAG) representing the data generation process.

The methods called meta-learners [17] such as S-Learner, T-learner, X-learner and R-learner are also designed to infer causal effect. They can be combined with any type of machine-learning algorithms including neural networks. There was an attempt to evaluate neural network-based meta-learners performance on the IHDP benchmark dataset [18], but until now it was unclear how such methods perform on other popular test datasets, namely, ACIC [19], and JOBS [20].

This review work complements two excellent and extensive overviews of causal inference methods by Yao et al. [4] and Koch et al. [21]. The former focused on categorizing existing causal inference methods. The latter discussed in detail the representation based neural networks as well as GAN based strategies and how they can be used for ITE estimation. In contrast to these reviews, our study gives an overview on how neural network methods for ITE estimations perform on various benchmark data sets. Further contributions of this article are presented below:

1) Publication of an extensive library of ITE estimation methods in TensorFlow that can be used for benchmarking or developing new methods.
2) Comprehensive benchmarking of meta-learners combined with neural networks on popular test datasets and comparison to other state-of-the art ITE estimation methods.
3) Improvement of several originally proposed models by varying the architecture of the main body and the treatment conditional branches, utilizing automatic hyperparameter optimization.

III. Methods

A. Problem Formulation

Suppose observational data $\boldsymbol{D} = \{[\boldsymbol{x}_i, y_i, t_i]\}_{i=1}^N$ consists of N subjects, where $\boldsymbol{x}_i \in \boldsymbol{X}$ with $\boldsymbol{X} \in \mathbb{R}^M$ is a set of M covariates, $t_i \in T$ with $T \in \{0, 1\}$ is the observed binary treatment, and $y_i \in Y$ with $Y \in \mathbb{R}$ is the factual outcome. Using the potential outcomes framework [22], let Y^1 be the potential outcome for subjects assigned to treatment group, and Y^0 be the potential outcome for people assigned to a control group. For simplicity, we assume that factual (observed), and counterfactual (unobserved) treatment outcomes are continuous. Since the results of both treatments are never observed simultaneously, we cannot calculate the individual treatment effect as $Y^1 - Y^0$. Instead, as in [23], we assume that there are no hidden confounders and that the data are independent of each other and estimate the Conditional Average Treatment Effect (CATE):

$$\begin{aligned} \text{CATE}(\boldsymbol{x}_i) &= \mathbb{E}[Y^1 - Y^0 | \boldsymbol{X} = \boldsymbol{x}_i] \\ &= \mathbb{E}[Y^1 | \boldsymbol{X} = \boldsymbol{x}_i] - \mathbb{E}[Y^0 | \boldsymbol{X} = \boldsymbol{x}_i] \\ &= \tau(\boldsymbol{x}_i). \end{aligned} \tag{1}$$

To evaluate the outcome prediction on a synthetic or semisynthetic dataset with ground-truth ITE denoted as $\hat{\tau}(\boldsymbol{x}_i)$ available for each individual, we compute the Precision in Estimating Heterogeneous Effect (PEHE) as:

$$\epsilon_{\text{PEHE}} = \frac{1}{N} \sum_{i=0}^{N} (\tau(\boldsymbol{x}_i) - \hat{\tau}(\boldsymbol{x}_i))^2. \tag{2}$$

In case the factual outcomes are available for the training set, but the test outcomes Y^0_{RCT} and Y^1_{RCT} are from the RCT, we can compute the policy risk [24] with:

$$\begin{aligned} \mathcal{R}_{pol} = 1 &- \mathbb{E}[Y^1 | \pi(\boldsymbol{X}) = 1] P(\pi(\boldsymbol{X}) = 1) \\ &+ \mathbb{E}[Y^0 | \pi(\boldsymbol{X}) = 0] P(\pi(\boldsymbol{X}) = 0), \end{aligned} \tag{3}$$

where $\pi(\boldsymbol{X}) = 1$ if $Y^1_{RCT} - Y^0_{RCT} > 0$ and $\pi(\boldsymbol{X}) = 0$, in the other way.

B. Meta-learners

We have chosen to implement meta-learners [17] as they are often used as building blocks of more advanced models and present basic architectural concepts to estimate counterfactual outcomes from treatment conditional input data. Meta-learners are strategies that can be combined with any regression or classification method. We include them in the review in order to show how they work with deep learning methods.

Single learner or S-Learner employ a single estimation function $\mu(\cdot)$, in our case a fully connected neural network, to predict the counterfactual outcomes. During the training, S-Learner receives as input covariates $\boldsymbol{X}$ concatenated with an observed treatments T. In the case of continuous outcomes the neural network weights are then updated to minimize the Mean Squared Error (MSE) between prediction and ground truth values Y:

$$\mathcal{L}_S = \mathbb{E}[(\mu(\boldsymbol{X}, T) - Y)^2] \tag{4}$$

During the inference, causal effect is computed as $\tau(\boldsymbol{x}_i) = \mu(\boldsymbol{x}_i, 1) - \mu(\boldsymbol{x}_i, 0))$ where we set the treatments to be one or zero respectively.

T-learner estimates response surfaces for each unique treatment value. In binary case, two causal estimators $\mu_0(\boldsymbol{x}_i) = \mathbb{E}[Y^0|\boldsymbol{X} = \boldsymbol{x}_i, t_i = 0]$ and $\mu_1(\boldsymbol{x}_i) = \mathbb{E}[Y^1|\boldsymbol{X} = \boldsymbol{x}_i, t_i = 1]$ are trained on the treatment specific covariates and outcomes [17]. The loss of the T-Learner is given by:

$$\mathcal{L}_T = \mathbb{E}[(1-T)(\mu_0(\boldsymbol{X}) - Y))^2 + T(\mu_1(\boldsymbol{X}) - Y)^2] \quad (5)$$

Like T-Learner, X-Learner first estimates response functions $\mu_0(\boldsymbol{x}_i)$ and $\mu_1(\boldsymbol{x}_i)$. After that, the imputed treatment effects are computed as: $D_i^0 := \mu_1(\boldsymbol{x}_i^0) - y_i^0$ and $D_i^1 := y_i^1 - \mu_0(\boldsymbol{x}_i^1)$, where $\boldsymbol{x}_i^1$, $\boldsymbol{x}_i^0$ and y_i^1, y_i^0 are observed covariates and outcomes for treated and untreated. Next, one computes $\tau_j(\boldsymbol{x}_i) = \mathbb{E}[D^j|\boldsymbol{X} = \boldsymbol{x}_i]$ with $j \in \{0, 1\}$ using machine learning models. Then CATE is inferred from $\tau(\boldsymbol{X}) = g(\boldsymbol{X})\tau_0(\boldsymbol{X}) + (1 - g(\boldsymbol{X}))\tau_1(\boldsymbol{X})$, in which the propensity score $g(\boldsymbol{X})$ is the treatment probability for a given set of covariates.

The R-Learner [25] also learns to estimate $\mu(\boldsymbol{x}_i)$ and $g(\boldsymbol{x}_i)$ with machine learning methods. The main difference is that the CATE estimator $\tau(\boldsymbol{x}_i)$ is additionally trained using neural networks. The loss is given by:

$$\mathcal{L}_R = \mathbb{E}[(Y - \mu(\boldsymbol{X}) - (T - g(\boldsymbol{X})\tau(\boldsymbol{X}))^2] \quad (6)$$

The disadvantage of S-Learner is that in the case of a multidimensional covariance space, the distinct role of the treatment variable T can be neglected, since the treatment variable is equated to $\boldsymbol{x}_i$ as an additional covariate. This can lead to a zero bias in treatment weights, leading to treatment assignments being ignored [17]. T-Learner is prone to loss of efficiency due to increased variance, since the data is grouped and processed strictly independently of each other, which prevents processing-independent representation in the latent space. X-learner relies on multiple estimators, moreover imputing the counterfactual variables inevitably increases the chance of accumulating errors. Representation based methods allow addressing the shortcomings of the aforementioned meta-learners.

C. Representation Based Methods

Shalit et al. [8] proposed a representation learning algorithm that reduces the distribution mismatch in the latent representation space called Counterfactual Regression (CFR), removing the bias caused by the imbalance between treatment and control groups. This method shows promising results, and is often considered cutting edge in this field.

Unlike meta-learners, it employs a hybrid architecture where all covariates are used as input. The body of the network consists of representation layers agnostic to treatment, followed by two treatment conditional branches or heads. Each head is only trained with samples matching the observed treatment. For binary treatment, the loss of a CFR model consists of two parts. The first part coincides with the loss of T-Learner (5), where $\mu_0(\cdot)$ and $\mu_1(\cdot)$ are the outcomes of treatment conditional branches. The second part is an additional loss term minimizing a distance based Integral Probability Metric

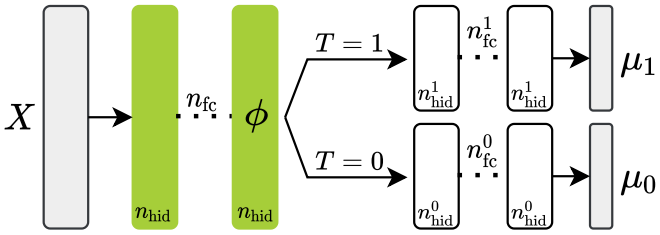

Fig. 1: The TAR-Net architecture by [8]. We added independent tunable hyperparameters for number of hidden units and fully connected layers respectively in the treatment conditional heads.

(IPM) of the two latent distributions $\phi(\boldsymbol{X}|T = 0)$ and $\phi(\boldsymbol{X}|T = 1)$. The chosen IPMs include the Wasserstein Metric [26], [27], and the Mean Maximum Discrepancy (MMD) [28]. We denote the corresponding methods as CFR-Wass and CFR-MMD. The total loss of the CFR model is presented below:

$$\mathcal{L}_{\text{CFR}} = \mathcal{L}_T + \alpha\,\text{IPM}(\phi(\boldsymbol{X}|T = 0), \phi(\boldsymbol{X}|T = 1)), \quad (7)$$

where $\alpha > 0$ is a group balance regularization. A variant of CFR with $\alpha = 0$ is called a Treatment-Agnostic Representation Network (TAR-Net) (see figure 1).

Instead of tackling group imbalance, Shi et al. [9] proposed the Dragon-Net architecture to adjust the TAR-Net architecture through an extra head for estimating the individual propensity scores $g(\boldsymbol{X})$. The underlying assumption is that the treatment effect is independent of the covariates $\boldsymbol{x}_i$ that are solely relevant for predicting outcome, but not the treatment. However, to prevent a direct propagation of estimation errors of $g(\boldsymbol{X})$ into the outcome model, the Dragon-Net learns an advantageous trade-off between predictive accuracy and the propensity-score representation. The objective is adjusted by the weighted Cross Entropy (CE) [29] loss between $g(\boldsymbol{X})$ and the actual treatment assignment, with $\alpha > 0$ being a hyperparameter:

$$\mathcal{L}_{\text{DN}} = \mathcal{L}_T + \alpha\,\text{CE}(g(\boldsymbol{X}), T), \quad (8)$$

Having a similar architecture to Dragon-Net, the third IPM based approach uses a propensity score to weight the impact of each conditional treatment branch during IPM calculation. According to [30], the introduction of these weightings provides more consistent guarantees in the event of significant disparities in treatment assignment. We refer to the weighted IPM method as CFR-Weight. As stated by [24] enforcing distribution equality with IPMs can be prone to information loss. Thus, the authors introduced Deep Kernel Learning ITE (DKLITE), a Bayesian model approach which optimizes the treatment effect estimation for minimum counterfactual variance by defining the upper bound of the ITE loss through the negative model likelihood and the posterior counterfactual variance. DKLITE, unlike CFR methods, does not use IPMs to ensure the balanced representation of treated and untreated subjects. The authors argue that enforcing domain invariance, i.e. equality between densities in latent representations of treated and control group, is unnecessary and can even be harmful in a high-dimensional space with a limited number of observations. Instead, they suggest learning representations

that cluster counterfactual data around representation of factual data, thus adjusting for the covariate shift. The algorithm first transforms the input through a fully connected neural network ϕ into the hidden space. After that, the result is passed through a kernel function. Next, the mean and variance of the hidden space distribution are calculated and are used for so-called variance and likelihood losses $\mathcal{L}_{\text{var}}$ and $\mathcal{L}_{\text{like}}$ respectively. Additionally, reconstruction loss $\mathcal{L}_{\text{rec}}$ is computed as MSE between input data $\boldsymbol{X}$ and the outcome of a network ϕ^{-1}. The final loss is given below:

$$\mathcal{L}_{\text{DKLITE}} = \mathcal{L}_{\text{like}} + \alpha_1 \mathcal{L}_{\text{var}} + \alpha_2 \mathcal{L}_{\text{rec}}, \tag{9}$$

where $\alpha_1 > 0$ and $\alpha_2 > 0$ are hyperparameters.

D. Adversarial Methods

Adversarial learning methods, in which two networks are simultaneously trained to compete against each other through coupling their loss objectives [15], can also be applied to the problem of ITE estimation. A typical Generative Adversarial Net (GAN) consists of generator and discriminator networks. The purpose of the generator network is to create samples as if they came from the target distribution. The discriminator network is trained to distinguish generated samples from real ones. The better the generated samples, the harder it is to distinguish them from the real ones for the discriminator network.

Yoon et al. [12] proposed to account for unobserved data by utilizing the GAN framework. The method GANITE (Generative Adversarial Nets for inference of Individualized Treatment Effects) developed by them consists of two blocks. The goal of the first block also called counterfactual block is to impute the missing counterfactual information using covariates as well as treatment and factual outcomes as input. The generator $G_{CF}(\boldsymbol{X}, Y, T)$ creates outcomes $\tilde{Y} = \{\tilde{Y}^0, \tilde{Y}^1\}$. The discriminator $D_{CF}(\boldsymbol{X}, \bar{Y})$, where $\bar{Y} = \{Y, \tilde{Y}\}$ is a vector of factual and predicted by generator outcomes, is trained to maximize the probability of detecting the factual outcomes in $\bar{Y}$, whereas the goal of the generator is to fool the discriminator by creating predictions similar to real ones. The losses for training discriminator and generator in counterfactual block are presented below:

$$\begin{aligned} \mathcal{L}_D &= -\mathcal{L}(D_{CF}) \\ \mathcal{L}_G &= \text{MSE}(Y, \tilde{Y}) + \alpha \mathcal{L}(D_{CF}), \end{aligned} \tag{10}$$

where $\alpha > 0$ is a hyperparameter. This results in a complete set of both factual and counterfactual outcomes, which is furthermore employed for training the second so called ITE-block. The goal of the ITE-block is to take information from the counterfactual block and use it as a guide to predict the outcomes solely based on the covariates. The ITE-block uses a generator $G_{ITE}(\boldsymbol{X})$ to create counterfactual outcomes $\hat{Y} = \{\hat{Y}^0, \hat{Y}^1\}$ based on the covariates. In the original paper, the quality of the prediction is then additionally verified by a discriminator trained to distinguish ITE after prediction with counterfactual block and ITE-block, but we omit it as a for the purpose of this paper an ITE discriminator is not needed. Loss for the ITE-block is presented below:

$$\mathcal{L}_{ITE} = \text{MSE}(\hat{Y}^1 - \hat{Y}^0, \bar{Y}^1 - \bar{Y}^0). \tag{11}$$

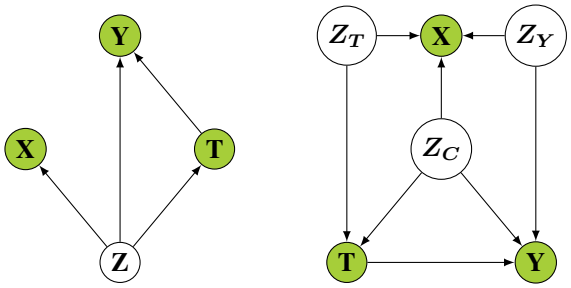

Fig. 2: DAG of CEVAE (left) and TEDVAE (right)

E. Variational Methods

Another approach is to learn latent variables, i.e. hidden confounders, through input covariates using variational autoencoders (VAE). As a probabilistic graphical model of Bayesian character, VAE approximates the observed distribution $p(\boldsymbol{X}|\boldsymbol{Z})$ (decoder) conditioned on latent variables $\boldsymbol{Z}$ sampled from the latent posterior distribution $q(\boldsymbol{Z}|\boldsymbol{X})$ (encoder). Both the decoder and encoder are simultaneously trained to maximize the evidence lower bound [31]. In the context of causal inference problem, VAE are adapted to DAGs in Figure 2, defining the process by which observations are drawn. Louizos et al. [11] proposed a Causal Effect Variational Autoencoder (CEVAE) that samples the proxy covariate distribution $p(\boldsymbol{X}|\boldsymbol{Z})$, binary treatment distribution $p(T|\boldsymbol{Z})$ and the outcome $p(Y|T, \boldsymbol{Z})$ from hidden variables. The inference network learns the posterior approximation through the complete input set $q(\boldsymbol{Z}|\boldsymbol{X}, Y, T)$. The overall training objective is determined through the variational lower bound of the model with the addition of auxiliary distributions $q(T|\boldsymbol{X})$ and $q(Y|\boldsymbol{X}, T)$. The loss of CEVAE is presented below:

$$\begin{aligned} \mathcal{L}_{\text{CEVAE}} = \mathbb{E}_{q(\boldsymbol{Z}|\boldsymbol{X},Y,T)}[&\log p(\boldsymbol{X}, T|\boldsymbol{Z}) + \log p(Y|T, \boldsymbol{Z}) \\ &+ \log p(\boldsymbol{Z}) - \log q(\boldsymbol{Z}|\boldsymbol{X}, Y, T)] \\ &+ \log q(T|\boldsymbol{X}) + \log q(Y|\boldsymbol{X}, T) \end{aligned} \tag{12}$$

Zhang et al. [13] adapted the ideas of CEVAE and proposed a model called Treatment Effect Disentangled Variational AutoEncoder (TEDVAE). Unlike CEVAE, which learns the combined latent representation to infer $\boldsymbol{X}, Y$, and T, TEDVAE separates the latent factors into three independent factors: Z_T, Z_Y, and Z_C. The instrumental factor Z_T affects only the treatment prescription, Z_Y only affects the outcome, and Z_C is a confounding factor affecting both treatment and outcome. Each disentangled factor is not a single value, but a distribution learned by separate encoders $q_T(\boldsymbol{Z}_T|\boldsymbol{X})$, $q_C(\boldsymbol{Z}_C|\boldsymbol{X})$ and $q_Y(\boldsymbol{Z}_Y|\boldsymbol{X})$. The parameters for each distribution come from fully connected neural networks.

The TEDVAE inference model consists of a decoder $p_X(\boldsymbol{X}|\boldsymbol{Z}_T, \boldsymbol{Z}_C, \boldsymbol{Z}_Y)$ reconstructing $\boldsymbol{X}$, two disjoint decoders $p_Y(Y|T = 1, \boldsymbol{Z}_C, \boldsymbol{Z}_Y)$, $p_Y(Y|T = 0, \boldsymbol{Z}_C, \boldsymbol{Z}_Y)$ predicting counterfactual outcomes, and $p_T(T|\boldsymbol{Z}_T, \boldsymbol{Z}_C)$ recovering the assigned treatment. The loss function is given by:

$$\begin{aligned} \mathcal{L}_{\text{TEDVAE}} = &\mathcal{L}_{\text{ELBO}}(\boldsymbol{X}, Y, T) \\ &+ \alpha_T \, \mathbb{E}_{q_T q_C}[\log p_T(T|\boldsymbol{Z}_T, \boldsymbol{Z}_C)] \\ &+ \alpha_Y \, \mathbb{E}_{q_Y q_C}[\log p_Y(Y|T, \boldsymbol{Z}_Y, \boldsymbol{Z}_C)], \end{aligned} \tag{13}$$

where $\alpha_T > 0$, $\alpha_Y > 0$ are hyperparameters, and $\mathcal{L}_{\text{ELBO}}(\boldsymbol{X}, Y, T)$ is:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}} = &\mathbb{E}_{q_T q_C q_Y}[\log p_X(\boldsymbol{X}|\boldsymbol{Z}_T, \boldsymbol{Z}_C, \boldsymbol{Z}_Y)] \\ &- D_{\text{KL}}(q_T(\boldsymbol{Z}_T|\boldsymbol{X})||p_T(\boldsymbol{Z}_T)) \\ &- D_{\text{KL}}(q_C(\boldsymbol{Z}_C|\boldsymbol{X})||p_C(\boldsymbol{Z}_C)) \\ &- D_{\text{KL}}(q_Y(\boldsymbol{Z}_Y|\boldsymbol{X})||p_Y(\boldsymbol{Z}_Y)).\end{aligned} \tag{14}$$

Here $q_T(\boldsymbol{Z}_T|\boldsymbol{X})$, $q_C(\boldsymbol{Z}_C|\boldsymbol{X})$ and $q_Y(\boldsymbol{Z}_Y|\boldsymbol{X})$ are Gaussian or Bernoulli distributions depending on a binary or continuous outcome variable, for which the mean and variance are parameterized by neural networks. Priors $p_T(\boldsymbol{Z}_T)$, $p_C(\boldsymbol{Z}_C)$, and $p_Y(\boldsymbol{Z}_Y)$ are represented by Gaussian normal distributions, and D_{KL} is a Kullback-Liebler divergence (KL) between them.

IV. Experiments

Due to the lack of datasets with ground truth treatment effect, in order to evaluate the model performance, we use open source semisynthetic benchmark datasets. We select datasets from the corresponding articles of methods reviewed in the previous section. The multiplicity of datasets from different sources allows for a broad comparison of models in terms of dimensionality, modeling counterfactual outcomes, the presence of hidden confounding factors, and proxy variables.

IHDP is based on a randomized control trial by Brooks-Gunn et al. [32] as part of the Infant Health and Development Program. It encompasses 747 instances, each of which contains 25 covariate variables that determine the characteristics of preterm infants with significantly low birth weight and their environment, such as information about parents. Treatment is an intensive child care program that includes home visits by physicians and specialists for a predetermined period of time. Counterfactual outcomes are randomly generated across predefined response surfaces using probabilistic models. For this work, the predefined modeling of counterfactual outcomes is similar to the procedure of Hill et al. [33], in which setting "A" generates a result that depends linearly on covariates, and setting "B" is a non-linear model, since the result of the control group is determined through an exponential function of covariates. We denote the datasets created using settings "A" and "B" as IHDP_a and IHDP_b respectively. For both settings, approximately 18% of the samples belong to the treatment group, indicating a significant class imbalance. We use 100 simulated IHDP datasets for each setting.

The **JOBS** dataset [20] is a combination of a National Supported Work Program randomized control trial and an observational study. The data consists of 3212 cases described by 8 covariates that define demographics and financial income in 1974 and 1975. Subjects assigned to the treated group undergo special professional training. The outcome under study is employment status. The included subgroup from RCT enables evaluation of the causal effect of "ground truth". The problem was first described by Shalit et al. [8] and is adopted accordingly. Among the considered datasets, JOBS shows the highest imbalance between treatment and control, with only 10% of the samples belonging to the treatment group. We create 100 train/test splits of the dataset to evaluate the model performance.

Dataset **ACIC** was published in the Atlantic Causal Inference Conference 2016 [19]. It is derived from linked birth and infant mortality data [34]. The dataset is based on IHDP data and contains 4802 observations with 58 covariates each. With a fixed set of observations, the dataset contains 77 simulated subsets of data with varying degrees of confidence in the correlation between treatment prescription, actual and counterfactual outcome, and non-linearity of treatment effect. For all subsets, the sample fraction of treatment groups is about 30%. It should be noted that overlap violations occur in the dataset, i.e. propensity score can reach extreme values close to 0 or 1 for certain covariates.

The methods described in Section III are considered in this work. While this only covers a subset of all treatment evaluation models using neural networks, we have focused on established methods with available code repositories. Thus, the goal of this work is not to create a complete library of all treatment evaluation models, but to list and quantify the differences between the established ITE estimation strategies and to encourage readers to use the results of our public library for their own research.

A. Implementation Details

For IHDP_a, IHDP_b, and JOBS datasets, all models were tuned on the first sub-dataset using the Random Search Tuner by Keras [35] as well as a TensorFlow callback function with *EarlyStopping* and *ReduceLROnPlateau*. Since ACIC encompasses 77 different datasets, each containing various numbers of sub-datasets, 77 models were tuned on the first data file of each of the ACIC sub-datasets.

Due to the large number of models and different datasets, direct hyperparameter optimization, namely searching on a fixed set of parameters and training the resulting models on the entire dataset, as done by [12] was not feasible. Moreover, it often showed worse performance compared to the Keras tuner Random Search strategy, since only individual sets of hyperparameters were taken into account. Any optimization algorithm can be prone to converging into a local minimum, so it is critical to evaluate different ranges of parameter values.

The models were tuned on a set of specified parameters as well as on architectural features such as the number of layers and nodes per layer. Further modifications of training parameters, e.g. kernel initializer and patience of *Keras callbacks*, were investigated.

Hypertuning was utilized to find an optimized set of hyperparameters which includes the number of layers, number of nodes, batch size as well as learning rate. In addition, we fixed the number of tuning epochs to 50 for all models and used a validation split value of 0.2. Validation loss was used as an early stop criteria during tuning to prevent model overfitting. For meta-learners and GANITE each sub-model was tuned separately. In particular, for GANITE, the learning rate was fixed for the generator and discriminator model. Likewise for representation based models such as TAR-Net, CFR-Wass and CFR-MMD, we tuned the number of hidden

TABLE I: Results and their 95% confidence intervals of each model on *IHDP* test dataset. The results of best performing models are marked in bold.

Model	$\text{IHDP}_\text{a}(\sqrt{\epsilon_\text{PEHE}})$	$\text{IHDP}_\text{b}(\sqrt{\epsilon_\text{PEHE}})$
S-Learner	0.41 ± 0.05	2.24 ± 0.06
T-Learner	0.51 ± 0.04	2.10 ± 0.06
R-Learner	0.68 ± 0.06	2.19 ± 0.05
X-Learner	0.75 ± 0.06	2.49 ± 0.08
TAR-Net	0.37 ± 0.02	$\mathbf{1.96 \pm 0.05}$
Dragon-Net	0.53 ± 0.03	2.02 ± 0.05
CFR-MMD	0.40 ± 0.06	2.06 ± 0.06
CFR-Wass	$\mathbf{0.36 \pm 0.04}$	2.10 ± 0.06
CFR-Weight	0.45 ± 0.05	1.97 ± 0.06
DKLITE	0.37 ± 0.03	2.11 ± 0.06
CEVAE	0.89 ± 0.10	2.81 ± 0.07
TEDVAE	0.54 ± 0.06	2.28 ± 0.07
GANITE	0.49 ± 0.04	2.27 ± 0.07

TABLE II: Results and their 95% confidence intervals of each model on *ACIC* and *JOBS* test datasets. The results of best performing models are marked in bold.

Model	ACIC $(\sqrt{\epsilon_\text{PEHE}})$	JOBS $(R_{Pol.})$
S-Learner	2.29 ± 0.09	0.23 ± 0.01
T-Learner	2.83 ± 0.09	0.25 ± 0.01
R-Learner	2.33 ± 0.09	0.24 ± 0.02
X-Learner	2.34 ± 0.10	0.24 ± 0.02
TAR-Net	2.38 ± 0.08	$\mathbf{0.21 \pm 0.01}$
Dragon-Net	2.81 ± 0.09	0.24 ± 0.01
CFR-MMD	2.43 ± 0.08	0.24 ± 0.02
CFR-Wass	2.68 ± 0.08	0.23 ± 0.01
CFR-Weight	2.60 ± 0.08	0.23 ± 0.01
DKLITE	$\mathbf{2.28 \pm 0.08}$	0.24 ± 0.02
CEVAE	2.89 ± 0.10	0.27 ± 0.02
TEDVAE	2.38 ± 0.08	0.24 ± 0.02
GANITE	2.34 ± 0.09	0.23 ± 0.01

layers and units for the main body and each of the treatment heads (see Figure 1). For CFR-Weight, we additionally tuned the parameters of propensity branch. For DKLITE, again, the encoder and decoder components were tuned independently on four hyperparameters. The latent dimension of the presentation space was tuned as well.

In the case of variational methods, we optimized the number of hidden layers and units in encoders and decoders. In addition, for CEVAE, the same two parameters were configured separately for the architectural parts encoding the covariance, treatment, and distribution of results in the inference network.

After finding the correct hyperparameters, the model was trained without using a validation split. During training, the learning rate was reduced when reaching a plateau for the training loss. We note that during the tuning process test dataset was not used.

B. Results

The results of trained models are presented in tables I and II. To ensure reproducibility, hyperparameters as well as implementation details of the tuned models are reported in https://github.com/causal-lab-miism/deep_ite_library.

The considered models perform differently on the binary response dataset $JOBS$ as well as on the regression problems given by the IHDP_a, IHDP_b and ACIC datasets.

S-Learner handles IHDP_a, ACIC and JOBS well, which indicates high inference ability in case of a simple dependency between covariates and treatment outcomes. This is confirmed by the relatively low performance on IHDP_b, modeled using the exponential response function. In contrast, T-Learner shows good ITE estimation capabilities for complex mappings, as it performs comparatively well on IHDP_b, but outputs inferior results for binary, linear, or versatile datasets. Although R-Learner and X-Leaner give acceptable results on ACIC, S-Learner performs better than both of them on IHDP_a. However, R-Learner outperforms S-Learner on IHDP_b indicating an advantage of the R-Learner in inferring ITE for highly non-linear data exclusively.

Among the representation based methods, TAR-Net, CFR-Wass and DKLITE showed similar and best performances on the IHDP_a dataset across all methods, even partially outperforming the model results presented in the original papers. TAR-Net and CFR-Weight give slightly better results on IHDP_b than the other methods, highlighting the versatility of the representational based methods for ITE estimation. CFR-MMD shows a PEHE performance deficit for IHDP_a compared to CFR-Wass, but has a slightly better PEHE for IHDP_b and a noticeable advantage for ACIC dataset. The same reasoning holds for the CFR-Weight model. As for the variational group, TEDVAE demonstrated superiority over the CEVAE method, which showed one of the lowest performances among the tested models on all presented datasets. Finally, GANITE showed good performance for JOBS, was ranked average compared to the other models for IHDP_a as well as IHDP_b and performed well for the ACIC dataset.

V. Discussion

Our results show that a simple multilayer perceptron in the form of an S-Learner model is able to achieve noticeable performance without a conditional treatment outcome assessment built into the architecture. In comparison, splitting treatment and control outcomes by employing two learners as implemented by T-Learner shows inferior performance on all datasets compared to S-Learner, which takes all covariates, including the treatment variable, as input data. This confirms the deterioration of T-Learner due to the increase in outcome variance because of the separation of input data into treatment and control samples, especially when there is a large imbalance between them.

Adapting multiple estimators, as done in R-Learner, shows similar behavior, although more independent estimators may lack the ability to generalize. Other meta-learners, performed better compared to X-Learner, which may be due to two non-exclusive reasons. X-Learner includes training of five networks, which firstly intensifies the error accumulation in estimates and secondly prevents convergence to the global minimum in both hyperparameter tuning and training. This leads to the conclusion that for low-dimensional data, the performance of advanced meta-learners is significantly degraded due to increased variance, accumulated errors, and a more complex optimization landscape.

The treatment-agnostic TAR-Net approach shows state-of-the-art and consistent counterfactual estimation performance compared to most other models, demonstrating the benefit of a shared hidden representation space for both treatment groups. The results of DKLITE further indicate that correctly adjusting the differences in the covariate distribution of treated and untreated subjects can help to estimate the counterfactual outcome more precisely in a variety of cases, such as represented by ACIC. The results of CFR models across all datasets, especially ACIC and JOBS, suggest that, compared to TAR-Net, forced overlapping of latent representation distributions for both the treatment and control group might be beneficial but as stated by Zhang et. al [24] it can lead to an increase in the complexity of finding optimal hyperparameters, resulting in a inferior performance. Given the results, the Dragon-Net architecture does not benefit from including a propensity estimate in the loss function; on the contrary, it limits the accurate estimation of ITE compared to TAR-Net. A possible reason for this could be the fact that the treatment probability is not determined, i.e. not derived from covariates for most datasets. As a consequence, the network is cluttered with redundant noise information that degrades hyperparameter tuning and training. Similar observations can be made when comparing the performance of the CFR-Weight model with the results of TAR-Net on some datasets. This leads to the conclusion that the correct choice of IPM is highly dataset dependent and should be explored independently for any given case.

The poor results of CEVAE may point to a complex landscape of its hyperparameters. Another possible reason for the reported results could be due to increased number of consecutively estimated distributions in the inference network compared to TEDVAE, which introduces more statistical error. However, TEDVAE by itself was unable to consistently achieve noticeable PEHE results on regression problem datasets, which again suggests that either the variational autoencoder approach is difficult to optimize with respect to the best set of hyperparameters, or that estimating and sampling of the disentangled conditional posterior distributions is an inferior approach compared to i.e. representation based methods.

GANITE achieved comparable results with other methods, justifying the generative approach for ITE. In addition, the potential of generative models for ITE estimation may not be fully exploited, since only one model was taken into account.

It is important to note that better results do not prove overall superiority over other models, but indicate a tendency to make consistent ITE estimates for a given dataset structure under automatic hyperparameter tuning. In addition, higher performance implies reduced hyperparameter optimization complexity.

VI. Conclusions

In this study, we reviewed deep learning methods within the task of assessing ITE and discussed their advantages and disadvantages, given the difficulty of tuning to optimal architectural and training parameters, as well as the ability to provide accurate ITE estimates. We applied the methods to different datasets and compared their performance under random search hyperparameter tuning strategy. Finally, we created an open source causal inference library written in TensorFlow to encourage readers to use the library for their own research or other applications in the field of causal estimation, and to use and adapt it for benchmarking purposes.

The results show that none of the methods in the group of meta-learners are generally superior to other methods. Among all the other methods, DKLITE, CFR-Nets and TAR-Net stood out because the algorithms were able to consistently infer state-of-the-art ITE scores on test datasets. This indicates a high level of generalizing ability of the models. However, it is important to note that other models can outperform TAR-Net and DKLITE on different datasets. For example, S-Learner performs very well on ACIC, leading to the suggestion that for various complex datasets, a simple architecture has the advantage of having a simplified tuning complexity.

Overall, we conclude that under automatic hyperparameter optimization, models of the representation based group outperform all other models for the current dataset selection, i.e. they have shown to better converge during tuning and training. Though the results are not sufficient to classify models in a dataset independent hierarchical sense, but rather to compare them under the task of tuning and training on the considered datasets, the results also indicate that automatic hyperparameter tuning strategy might lead to an enhanced ITE estimation ability of the models. The benefits provided by hyperparameter optimization may encourage researchers to focus on this aspect when creating new methods for ITE estimation.

For future work, it might be interesting to further refine the representation based methods build on the TAR-net, CFR-Net or DKLITE approaches. Alternatively, the potential of generative models for ITE may not be fully exploited so far and therefore may also be the focus of future research. In addition, it would be interesting to combine the methods presented in our library into an automatic causal inference library similar to [36].

References

[1] D. B. Rubin, “Estimating causal effects of treatments in randomized and nonrandomized studies.” *Journal of Educational Psychology*, vol. 66, no. 5, pp. 688–701, 1974. [Online]. Available: http://content.apa.org/journals/edu/66/5/688

[2] M. Hernan and J. Robins, *Causal Inference*, ser. Chapman & Hall/CRC Monographs on Statistics & Applied Probab. Taylor & Francis, 2023. [Online]. Available: https://books.google.de/books?id=_KnHIAAACAAJ

[3] J. Pearl and J. M. Robins, "Probabilistic evaluation of sequential plans from causal models with hidden variables," in *UAI*, 1995.

[4] L. Yao, Z. Chu, S. Li, Y. Li, J. Gao, and A. Zhang, "A Survey on Causal Inference," *ACM Transactions on Knowledge Discovery from Data*, vol. 15, no. 5, pp. 1–46, Oct. 2021. [Online]. Available: https://dl.acm.org/doi/10.1145/3444944

[5] E. Karavani, P. Bak, and Y. Shimoni, "A discriminative approach for finding and characterizing positivity violations using decision trees," *arXiv:1907.08127 [cs, stat]*, Jul. 2019, arXiv: 1907.08127. [Online]. Available: http://arxiv.org/abs/1907.08127

[6] P. Hoyer, D. Janzing, J. M. Mooij, J. Peters, and B. Schölkopf, "Nonlinear causal discovery with additive noise models," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21. Curran Associates, Inc., 2008. [Online]. Available: https://proceedings.neurips.cc/paper/2008/file/f7664060cc52bc6f3d620bcedc94a4b6-Paper.pdf

[7] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0893608014002135

[8] U. Shalit, F. D. Johansson, and D. Sontag, "Estimating individual treatment effect: generalization bounds and algorithms," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 3076–3085.

[9] C. Shi, D. M. Blei, and V. Veitch, "Adapting Neural Networks for the Estimation of Treatment Effects," in *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.

[10] X. Du, L. Sun, W. Duivesteijn, A. Nikolaev, and M. Pechenizkiy, "Adversarial balancing-based representation learning for causal effect inference with observational data," *Data Mining and Knowledge Discovery*, vol. 35, no. 4, pp. 1713–1738, Jul. 2021. [Online]. Available: https://link.springer.com/10.1007/s10618-021-00759-3

[11] C. Louizos, U. Shalit, J. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal Effect Inference with Deep Latent-Variable Models," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 6449–6459, event-place: Long Beach, California, USA.

[12] J. Yoon, J. Jordon, and M. v. d. Schaar, "GANITE: Estimation of Individualized Treatment Effects using Generative Adversarial Nets," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=ByKWUeWA-

[13] W. Zhang, L. Liu, and J. Li, "Treatment effect estimation with disentangled latent factors," in *Proceedings of the Thirty-fifth AAAI Conference on Artificial Intelligence (AAAI'21)*, 2021.

[14] M. J. Vowels, N. C. Camgoz, and R. Bowden, "Targeted VAE: Variational and Targeted Learning for Causal Inference," in *2021 IEEE International Conference on Smart Data Services (SMDS)*. Chicago, IL, USA: IEEE, Sep. 2021, pp. 132–141. [Online]. Available: https://ieeexplore.ieee.org/document/9592419/

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020, place: New York, NY, USA Publisher: Association for Computing Machinery. [Online]. Available: https://doi.org/10.1145/3422622

[16] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv:1312.6114 [cs, stat]*, May 2014, arXiv: 1312.6114. [Online]. Available: http://arxiv.org/abs/1312.6114

[17] S. R. Künzel, J. S. Sekhon, P. J. Bickel, and B. Yu, "Metalearners for estimating heterogeneous treatment effects using machine learning," *Proceedings of the National Academy of Sciences*, vol. 116, no. 10, pp. 4156–4165, Mar. 2019. [Online]. Available: https://pnas.org/doi/full/10.1073/pnas.1804597116

[18] A. Curth and M. van der Schaar, "Nonparametric Estimation of Heterogeneous Treatment Effects: From Theory to Learning Algorithms," in *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Banerjee and K. Fukumizu, Eds., vol. 130. PMLR, Apr. 2021, pp. 1810–1818. [Online]. Available: https://proceedings.mlr.press/v130/curth21a.html

[19] V. Dorie, J. Hill, U. Shalit, M. Scott, and D. Cervone, "Automated versus Do-It-Yourself Methods for Causal Inference: Lessons Learned from a Data Analysis Competition," *Statistical Science*, vol. 34, no. 1, Feb. 2019. [Online]. Available: https://projecteuclid.org/journals/statistical-science/volume-34/issue-1/Automated-versus-Do-It-Yourself-Methods-for-Causal-Inference/10.1214/18-STS667.full

[20] R. H. Dehejia and S. Wahba, "Causal Effects in Nonexperimental Studies: Reevaluating the Evaluation of Training Programs," *Journal of the American Statistical Association*, vol. 94, no. 448, pp. 1053–1062, Dec. 1999. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/01621459.1999.10473858

[21] B. Koch, T. Sainburg, P. Geraldo, S. Jiang, Y. Sun, and J. G. Foster, "Deep Learning of Potential Outcomes," *arXiv:2110.04442 [cs, econ, stat]*, Oct. 2021, arXiv: 2110.04442. [Online]. Available: http://arxiv.org/abs/2110.04442

[22] D. B. Rubin, "Causal Inference Using Potential Outcomes: Design, Modeling, Decisions," *Journal of the American Statistical Association*, vol. 100, no. 469, pp. 322–331, Mar. 2005. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1198/016214504000001880

[23] A. Jesson, S. Mindermann, U. Shalit, and Y. Gal, "Identifying Causal-Effect Inference Failure with Uncertainty-Aware Models," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 11 637–11 649. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/860b37e28ec7ba614f00f9246949561d-Paper.pdf

[24] Y. Zhang, A. Bellot, and M. van der Schaar, "Learning Overlapping Representations for the Estimation of Individualized Treatment Effects," in *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, S. Chiappa and R. Calandra, Eds., vol. 108. PMLR, Aug. 2020, pp. 1005–1014. [Online]. Available: https://proceedings.mlr.press/v108/zhang20c.html

[25] X. Nie and S. Wager, "Quasi-oracle estimation of heterogeneous treatment effects," *Biometrika*, vol. 108, no. 2, pp. 299–319, Sep. 2020. [Online]. Available: https://doi.org/10.1093/biomet/asaa076

[26] C. Villani, *Optimal transport: old and new*, ser. Grundlehren der mathematischen Wissenschaften. Berlin: Springer, 2009, no. 338.

[27] M. Cuturi and A. Doucet, "Fast Computation of Wasserstein Barycenters," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32. Bejing, China: PMLR, Jun. 2014, pp. 685–693, issue: 2. [Online]. Available: https://proceedings.mlr.press/v32/cuturi14.html

[28] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A Kernel Two-Sample Test," *Journal of Machine Learning Research*, vol. 13, no. 25, pp. 723–773, 2012. [Online]. Available: http://jmlr.org/papers/v13/gretton12a.html

[29] G. Cybenko, D. O'Leary, and J. Rissanen, *The Mathematics of Information Coding, Extraction and Distribution*, ser. The IMA Volumes in Mathematics and its Applications. Springer New York, 1998. [Online]. Available: https://books.google.de/books?id=jDrp4QEGioMC

[30] F. D. Johansson, U. Shalit, N. Kallus, and D. Sontag, "Generalization Bounds and Representation Learning for Estimation of Potential Outcomes and Causal Effects," *arXiv:2001.07426 [cs, stat]*, Feb. 2022, arXiv: 2001.07426. [Online]. Available: http://arxiv.org/abs/2001.07426

[31] D. J. Im, K. Cho, and N. Razavian, "Causal Effect Variational Autoencoder with Uniform Treatment," *arXiv:2111.08656 [cs]*, Nov. 2021, arXiv: 2111.08656. [Online]. Available: http://arxiv.org/abs/2111.08656

[32] J. Brooks-Gunn, F. R. Liaw, and P. K. Klebanov, "Effects of early intervention on cognitive function of low birth weight preterm infants," *The Journal of Pediatrics*, vol. 120, no. 3, pp. 350–359, Mar. 1992.

[33] J. L. Hill, "Bayesian Nonparametric Modeling for Causal Inference," *Journal of Computational and Graphical Statistics*, vol. 20, no. 1, pp. 217–240, Jan. 2011. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1198/jcgs.2010.08162

[34] M. F. MacDorman and J. O. Atkinson, "Infant mortality statistics from the linked birth/infant death data set–1995 period data," *Monthly Vital Statistics Report*, vol. 46, no. 6 Suppl 2, pp. 1–22, Feb. 1998.

[35] T. O'Malley, E. Bursztein, J. Long, F. Chollet, H. Jin, L. Invernizzi *et al.*, "Kerastuner," https://github.com/keras-team/keras-tuner, 2019.

[36] A. Alaa and M. V. D. Schaar, "Validating Causal Inference Models via Influence Functions," in *Proceedings of the 36th International Conference on Machine Learning*. PMLR, May 2019, pp. 191–201. [Online]. Available: https://proceedings.mlr.press/v97/alaa19a.html