PERFECT MATCH: A SIMPLE METHOD FOR LEARNING REPRESENTATIONS FOR COUNTERFACTUAL INFERENCE WITH NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning representations for counterfactual inference from observational data is of high practical relevance for many domains, such as healthcare, public policy and economics. Counterfactual inference enables one to answer "What if...?" questions, such as "What would be the outcome if we gave this patient treatment t_1?". However, current methods for training neural networks for counterfactual inference on observational data are either overly complex, limited to settings with only two available treatment options, or both. Here, we present Perfect Match (PM), a method for training neural networks for counterfactual inference that is easy to implement, compatible with any architecture, does not add computational complexity or hyperparameters, and extends to any number of treatments. PM is based on the idea of augmenting samples within a minibatch with their propensity-matched nearest neighbours. Our experiments demonstrate that PM outperforms a number of more complex state-of-the-art methods in inferring counterfactual outcomes across several real-world and semi-synthetic datasets.

1 INTRODUCTION

Estimating individual treatment effects (ITE) from observational data is an important problem in many domains. In medicine, for example, we would be interested in using data of people that have been treated in the past to predict what medications would lead to better outcomes for new patients (Shalit et al. (2017)). Similarly, in economics, we would for example want to determine how effective job programs would be based on results of past job training programs (LaLonde (1986)).

ITE estimation from observational data is difficult for two reasons: Firstly, we never observe all potential outcomes. If a patient is given a treatment to treat her symptoms, we never observe what *would have happened if* the patient was prescribed a potential alternative treatment in the same situation. Secondly, the assignment of cases to treatments is typically biased such that cases for which a given treatment is more effective are more likely to have received that treatment. The distribution of samples may therefore differ significantly between the treated group and the overall population. A supervised model naïvely trained to minimise the factual error would overfit to the properties of the treated group, and thus not generalise well to the entire population.

To address these problems, we introduce Perfect Match (PM), a simple method for training neural networks for counterfactual inference that extends to any number of treatment options. PM effectively controls for biased assignment of treatments in observational data by augmenting every sample within a minibatch with its closest matches by propensity score from the other treatment options. PM is easy to use with existing neural network architectures, simple to implement, and does not add any hyperparameters or computational complexity. We perform experiments that demonstrate that PM outperforms a number of more complex state-of-the-art methods in inferring counterfactual outcomes on several real-world and semi-synthetic datasets. Our experiments also show that PM is more robust to a high level of treatment assignment bias than existing methods. We believe that our simple and effective method for ITE estimation that extends to any number of treatment options could potentially make counterfactual inference more accessible, particularly for medical applications, where multiple available treatment options are the norm.

Contributions. This work contains the following contributions:

- We introduce Perfect Match, a simple methodology for learning neural representations for counterfactual inference based on propensity score matching within minibatches.
- We perform extensive experiments on semi-synthetic data in both the binary and multiple treatment setting. The experiments show that PM outperforms a number of more complex state-of-the-art methods in inferring counterfactual outcomes.

2 RELATED WORK

There are five main categories of approaches to learning to estimate ITEs:

Matching-based Methods. Matching methods estimate the counterfactual outcome of a sample X with respect to treatment t using the factual outcomes of its nearest neighbours having received t, with respect to a metric space. k-Nearest-Neighbour (kNN) (Ho et al. (2007)) operates in the potentially high-dimensional covariate space, and therefore might suffer from the curse of dimensionality. Propensity Score Matching (PSM) (Rosenbaum & Rubin (1983)) matches on the scalar probability $p(t|X)$ of t given the covariates X. PSM may require under- or oversampling the original data.

Adjusted Regression Methods. Adjusted regression models apply regression models with both treatment and covariates as input. The simplest case is Ordinary Least Squares, which can either be used for building one model, with the treatment as an input feature, or multiple separate models, one for each treatment (Kallus (2017)). More complex regression models, such as Treatment-Agnostic Representation Networks (TARNET) (Shalit et al. (2017)) may be used to capture non-linear relationships. Methods that make use of propensity scores, and a model of the outcomes, such as Propensity Dropout (PD) (Alaa et al. (2017)), are referred to as doubly robust (Funk et al. (2011)).

Tree-based Methods. Tree-based methods train many weak learners to build expressive ensemble models. Examples of tree-based methods are Bayesian Additive Regression Trees (BART) (Chipman et al. (2010); Chipman & McCulloch (2016)) and Causal Forests (CF) (Wager & Athey (2017)).

Representation-balancing Methods. Representation-balancing methods seek to learn a high-level representation for which the covariate distributions are balanced across treatment groups. Balancing Neural Networks (Johansson et al. (2016)) attempt to find such representations by minimising the discrepancy distance (Mansour et al. (2009)) between treatment groups. Counterfactual Regression Networks (CFRNET, Shalit et al. (2017)) use different metrics such as the Wasserstein distance.

Distribution-modeling Methods. Generative Adversarial Nets for inference of Individualised Treatment Effects (GANITE) (Yoon et al. (2018)) address ITE estimation using counterfactual and ITE generators. GANITE uses a complex architecture with many hyperparameters and submodels that may be difficult to implement and optimise. Causal Effect Variational Autoencoders (CEVAEs) (Louizos et al. (2017)) use a latent variable modeling approach that is robust to hidden confounding, but does not address biased treatment assignment. Causal Multi-task Gaussian Processes (CMGP) (Alaa & van der Schaar (2017)) apply a multi-task Gaussian Process to ITE estimation. The optimisation of CMGPs involves a matrix inversion of $O(n^3)$ complexity that limits their scalability.

In contrast to existing methods, PM is a simple doubly robust method built on propensity score matching that can be used to train expressive non-linear neural network models for ITE estimation from observational data in settings with any number of treatments (Table 1). While the underlying idea behind PM is simple and effective, it has, to the best of our knowledge, not yet been explored.

Table 1: Comparison of several representative state-of-the-art methods for counterfactual inference showing whether they were designed for > 2 treatments and their complexity of implementation.

	PM	CF	BART	BNN	CFRNET	PD	CEVAE	CMGP	GANITE
> 2 Treatments	✓	✓	✓	×	×	×	×	×	✓
Complexity	Low	Low	Low	Low	Medium	Medium	High	High	High

3 METHODOLOGY

Problem Setting. We consider a setting in which we are given N observed samples X, where each sample consists of p covariates x_i with $i \in [0, p)$. For each sample, the potential outcomes are represented as a vector Y with k entries y_j where each entry corresponds to the outcome when applying one treatment t_j out of the set of k available treatments $T = \{t_0, ..., t_{k-1}\}$ with $j \in [0, k)$. As training data, we receive observed samples X and their factual outcomes y_j when applying one treatment t_j. The set of available treatments can contain two or more treatment options. We refer to the special case of two available treatment options as the binary treatment setting. Given the observational training data with factual outcomes, we wish to train a predictive model that is able to accurately produce a predicted potential outcomes vector $\hat{Y}$ with k entries $\hat{y}_j$. In literature, this setting is known as the Rubin-Neyman potential outcomes framework (Rubin (2005)).

Precision in Estimation of Heterogenous Effect (PEHE). The primary metric that we optimise for when training models to estimate ITE is the PEHE (Hill (2011)). In the binary setting, the PEHE measures the ability of a predictive model to estimate the difference in effect between two treatments t_0 and t_1 for samples X. To compute the PEHE, we measure the mean squared error between the true difference in effect $y_1(n) - y_0(n)$, drawn from the noiseless underlying outcome distributions μ_1 and μ_0, and the predicted difference in effect $\hat{y}_1(n) - \hat{y}_0(n)$ indexed by n over N samples:

$$\epsilon_{\text{PEHE}} = \frac{1}{N} \sum_{n=0}^{N} \Big(\mathbb{E}_{y_j(n) \sim \mu_j(n)} [y_1(n) - y_0(n)] - [\hat{y}_1(n) - \hat{y}_0(n)] \Big)^2 \tag{1}$$

When the underlying noiseless distributions μ_j are not known, the true difference in effect $y_1(n) - y_0(n)$ can be estimated using the noisy ground truth outcomes y_i (Appendix A). As a secondary metric, we consider the error ϵ_{ATE} in estimating the average treatment effect (ATE) (Hill (2011)). The ATE measures the average difference in effect across the whole population (Appendix B). The ATE is not as important as PEHE for models optimised for ITE estimation, but can be a useful indicator of how well an ITE estimator performs at comparing two treatments across the entire population. We can neither calculate ϵ_{PEHE} nor ϵ_{ATE} without knowing the outcome generating process.

Multiple Treatments. Both ϵ_{PEHE} and ϵ_{ATE} can be trivially extended to multiple treatments by considering the average PEHE and ATE between every possible pair of treatments. Note that we lose the information about the precision in estimating ITE between specific pairs of treatments by averaging over all $\binom{k}{2}$ pairs. However, one can inspect the pair-wise PEHE to get the whole picture.

$$\hat{\epsilon}_{\text{mPEHE}} = \frac{1}{\binom{k}{2}} \sum_{i=0}^{k-1} \sum_{j=0}^{i-1} \hat{\epsilon}_{\text{PEHE},i,j} \tag{2} \qquad \hat{\epsilon}_{\text{mATE}} = \frac{1}{\binom{k}{2}} \sum_{i=0}^{k-1} \sum_{j=0}^{i-1} \hat{\epsilon}_{\text{ATE},i,j} \tag{3}$$

Perfect Match (PM). We consider fully differentiable neural network models optimised via minibatch stochastic gradient descent to predict potential outcomes $\hat{Y}$ given a sample X. To address the treatment selection bias inherent in observational data, we propose the use of a training methodology based on propensity score matching. A propensity score is the conditional probability $p(t|X)$ of a given sample X receiving a specific treatment t (Rosenbaum & Rubin (1983); Ho et al. (2007)). The propensity score is a *balancing score*, meaning that if the distribution of propensity scores across treatment groups is the same, then the distribution of their covariates x_i is also guaranteed to be the same across treatment groups (Ho et al. (2007)). By matching samples with their nearest neighbours by propensity score we can, in the optimal case, balance the covariates x_i. This breaks the dependence of treatment assignment on X if all relevant covariates are observed, and therefore effectively removes any potential treatment assignment bias (Ho et al. (2007)). However, in practice, we do not have access to the true treatment assignment probability $p(t|X)$. We therefore have to estimate the propensity score $p(t|X)$ by training a predictive model to predict the likelihood of receiving a treatment t for a given sample X. Note that propensity scores are not the only balancing score (Ho et al. (2007)). For example, X itself is another balancing score. However, matching on the one-dimensional propensity score is preferable because it avoids the curse of dimensionality that would be associated with matching on the potentially high-dimensional X directly. In PM, we match every sample within a minibatch with its nearest neighbours by propensity score from all other treatments (Pseudocode in Appendix C), taken from the training set. Every minibatch the model is trained on

Table 2: Comparisons of the datasets used in our experiments. We evaluate on three semi-synthetic datasets with varying numbers of treatments and samples, and a real-world clinical trial dataset.

Dataset	Type	# Samples	# Features	# Treatments	Counterfactuals
IHDP	semi-synthetic	747	25	2	available
Jobs	real-world	3212	7	2	not available
News	semi-synthetic	5000	2870	2/4/8/16	available
TCGA	semi-synthetic	9659	20531	4	available

therefore contains the same number of samples for each treatment group, and the covariates x_i of each treatment group are approximately balanced on average. The intuition behind matching on the minibatch level, rather than the dataset level (Ho et al. (2011)), is that it ensures that every gradient step is done in a way that is approximately unbiased by the treatment assignment bias inherent in observational data. In this sense, PM can be seen as a minibatch sampling strategy (Csiba & Richtárik (2018)) specifically designed to improve learning for counterfactual inference.

Model Selection. Besides accounting for the treatment assignment bias, the other major issue in learning for counterfactual inference from observational data is that, given multiple models, it is not trivial to decide which one to select. The root problem is that we do not have direct access to the true error in estimating counterfactual outcomes, only the error in estimating the observed factual outcomes. This makes it difficult to perform parameter and hyperparameter optimisation, as we do not know which models are better than others for counterfactual inference on a given dataset. To rectify this problem, we use a nearest neighbour approximation $\hat{\epsilon}_{\text{NN-PEHE}}$ of the $\hat{\epsilon}_{\text{PEHE}}$ metric for the binary (Shalit et al. (2017)) and multiple treatment settings for model selection. The $\hat{\epsilon}_{\text{NN-PEHE}}$ estimates the treatment effect of a given sample by substituting the true counterfactual outcome with the outcome y_j from a respective nearest neighbour NN matched on X using the Euclidean distance.

$$\hat{\epsilon}_{\text{NN-PEHE}} = \frac{1}{N} \sum_{n=0}^{N} \Big([y_1(\text{NN}(n)) - y_0(\text{NN}(n))] - [\hat{y}_1(n) - \hat{y}_0(n)] \Big)^2 \tag{4}$$

Analogously to Equations (2) and (3), the $\hat{\epsilon}_{\text{NN-PEHE}}$ metric can be extended to the multiple treatment setting by considering the mean $\hat{\epsilon}_{\text{NN-PEHE}}$ between all $\binom{k}{2}$ possible pairs of treatments (Appendix D).

4 Experiments

Our experiments aimed to answer the following questions:

1. What is the comparative performance of PM in inferring counterfactual outcomes in the binary and multiple treatment setting compared to existing state-of-the-art methods?
2. Does model selection by NN-PEHE outperform selection by factual MSE?
3. How does the relative number of matched samples within a minibatch affect performance?
4. How well does PM cope with an increasing treatment assignment bias in the observed data?
5. How does the presence of hidden confounders influence the performance of PM?

4.1 Datasets

We performed experiments on four real-world and semi-synthetic datasets (Table 2) with binary and multiple treatment options in order to gain a better understanding of the empirical properties of PM.

Infant Health and Development Program (IHDP). The IHDP dataset (Hill (2011)) contains data from a randomised study on the impact of specialist visits on the cognitive development of children, and consists of 747 children with 25 covariates describing properties of the children and their mothers. Children that did not receive specialist visits were part of a control group. The outcomes were simulated using the NPCI package from Dorie (2016). The IHDP dataset is biased because the treatment groups had a biased subset of the treated population removed (Shalit et al. (2017)). We used the same simulated outcomes[1] as Shalit et al. (2017).

Jobs. The Jobs dataset (LaLonde (1986)) is a blend of data from randomised and observational studies on the effect of professional training programs on unemployment, and is a commonly used benchmark dataset for counterfactual inference in the binary setting. Each sample consisted of demographic covariates, such as age, gender and previous income, and the treatment was enrolment

[1]Available at: `http://www.mit.edu/~fredrikj/files/IHDP-1000.tar.gz`

in a job training program. All samples that were not enrolled in a training program were considered control cases. The task was to predict the potential unemployment given the covariates and treatment. We used the same feature and sample set as described in Dehejia & Wahba (2002); Smith & Todd (2005); Shalit et al. (2017). A subgroup of the data was randomised and we therefore were able to estimate the ground truth average treatment effect on the treated (ATT) (Shalit et al. (2017)). Treatment assignment in Jobs is biased because parts of the data come from an observational study.

News. The News dataset was first proposed as a benchmark for counterfactual inference by Johansson et al. (2016) and consists of 5000 randomly sampled news articles from the NY Times corpus[2]. The News dataset contains data on the opinion of media consumers on news items. The samples X represent news items consisting of word counts $x_i \in \mathbb{N}$, the outcome $y_j \in \mathbb{R}$ is the reader's opinion of the news item, and the k available treatment options represent various devices that could be used for viewing, e.g. smartphone, tablet, desktop, television or others (Johansson et al. (2016)). We extended the original dataset specification in (Johansson et al. (2016)) to enable the simulation of arbitrary numbers of viewing devices. To model that consumers prefer to read certain media items on specific viewing devices, we train a topic model on the whole NY Times corpus and define $z(X)$ as the topic distribution of news item X. We then randomly pick $k+1$ centroids in topic space, with k centroids z_j per viewing device and one control centroid z_c. We assigned a random Gaussian outcome distribution with mean $\mu_j \sim \mathcal{N}(0.45, 0.15)$ and standard deviation $\sigma_j \sim \mathcal{N}(0.1, 0.05)$ to each centroid. For each sample, we drew ideal potential outcomes from that Gaussian outcome distribution $\tilde{y}_j \sim \mathcal{N}(\mu_j, \sigma_j) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.15)$. We then defined the unscaled potential outcomes $\bar{y_j} = \tilde{y}_j * [\mathrm{D}(z(X), z_j) + \mathrm{D}(z(X), z_c)]$ as the ideal potential outcomes $\tilde{y}_j$ weighted by the sum of distances to centroids z_j and the control centroid z_c using the Euclidean distance as distance D. We assigned the observed treatment t using $t|x \sim \mathrm{Bern}(\mathrm{softmax}(\kappa \bar{y_j}))$ with a treatment assignment bias coefficient κ, and the true potential outcome $y_j = C\bar{y_j}$ as the unscaled potential outcomes $\bar{y_j}$ scaled by a coefficient $C = 50$. We used four different variants of this dataset with $k = 2$, 4, 8, and 16 viewing devices, and $\kappa = 10$, 10, 10, and 7, respectively. Higher values of κ indicate a higher expected treatment assignment bias depending on $\bar{y_j}$. $\kappa = 0$ indicates no assignment bias.

The Cancer Genomic Atlas (TCGA). The TCGA project collected gene expression data from various types of cancers in 9659 individuals (Weinstein et al. (2013)). There were four available clinical treatment options: (1) medication, (2) chemotherapy, (3) surgery, or (4) both surgery and chemotherapy. We used a synthetic outcome function that simulated the risk of cancer recurrence after receiving either of the treatment options based on the real-world gene expression data. Before further processing, we standardised the gene expression data using the mean and standard deviations of gene expression at each gene locus for normal tissue in the training set. To produce the potential outcomes Y, we first selected $k+1$ gene expression phenotypes as representative phenotypes: One phenotype X_j for each treatment option and one as a control phenotype X_c. Analogously to the News dataset, we assigned a random Gaussian outcome distribution with mean $\mu_j \sim \mathcal{N}(0.45, 0.15)$ and standard deviation $\sigma_t \sim \mathcal{N}(0.1, 0.05)$ to each phenotype. For each sample, we drew ideal po-

[2]https://archive.ics.uci.edu/ml/datasets/bag+of+words

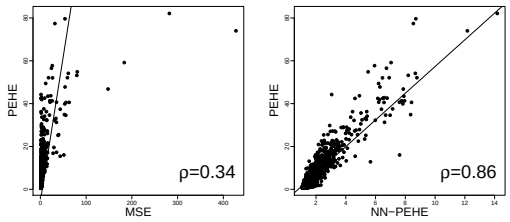


Figure 1: Correlation analysis of the real PEHE (y-axis) with the mean squared error (MSE; left) and the nearest neighbour approximation of the precision in estimation of heterogenous effect (NN-PEHE; right) across over 20'000 model evaluations on the validation set of IHDP. Scatterplots show a subsample of 1'400 data points. ρ indicates the Pearson correlation. NN-PEHE correlates significantly better with PEHE than MSE.

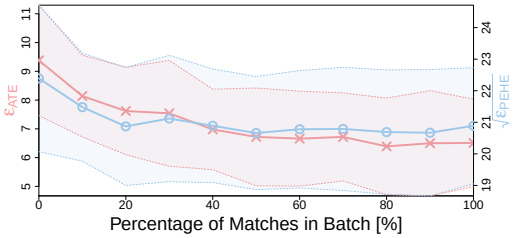


Figure 2: Change in error (y-axes) in terms of precision in estimation of heterogenous effect (PEHE) and average treatment effect (ATE) when increasing the percentage of matches in each minibatch (x-axis). Symbols correspond to the mean value of $\hat{\epsilon}_{\text{ATE}}$ (red) and $\sqrt{\hat{\epsilon}_{\text{PEHE}}}$ (blue) on the test set of News-8 across 50 repeated runs with new outcomes (lower is better). Performance improves with more matches added.

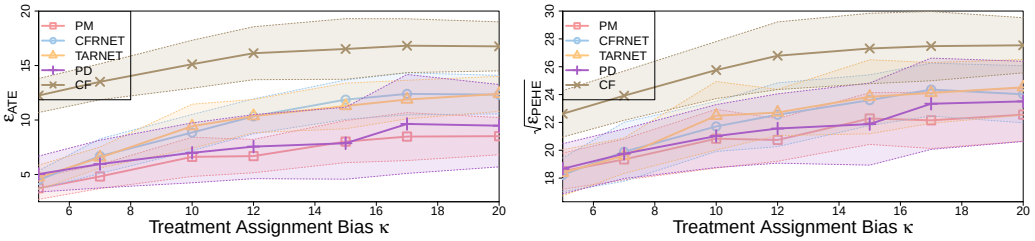


Figure 3: Comparison of several state-of-the-art methods for counterfactual inference on the test set of the News-8 dataset when varying the treatment assignment imbalance κ (x-axis), i.e. how much the treatment assignment is biased towards more effective treatments. Symbols correspond to the mean value of $\hat{\epsilon}_{\text{ATE}}$ (left) and $\sqrt{\hat{\epsilon}_{\text{PEHE}}}$ (right) across 50 repeated runs with new outcomes (lower is better). The shaded area indicates the standard deviation. PM handles an increasing treatment assignment bias κ better than existing state-of-the-art methods in terms of both $\hat{\epsilon}_{\text{ATE}}$ and $\sqrt{\hat{\epsilon}_{\text{PEHE}}}$.

tential outcomes from that Gaussian outcome distribution $\tilde{y}_j \sim \mathcal{N}(\mu_j, \sigma_j) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.15)$. We then defined the unscaled potential outcomes $\bar{y_j} = \tilde{y}_j * [\text{D}(X, X_{t_j}) + \text{D}(X, X_c)]$ as the ideal potential outcomes $\tilde{y}_j$ weighted by the sum of distances to phenotype X_{t_j} and the control phenotype X_c using the cosine similarity as distance metric D. We assigned the observed treatment t using $t|x \sim \text{Bern}(\text{softmax}(\kappa \bar{y_j}))$ with a treatment assignment bias coefficient $\kappa = 10$, and the true potential outcome $y_j = C\bar{y_j}$ as the unscaled potential outcomes $\bar{y_j}$ scaled by a coefficient $C = 50$.

All datasets with the exception of IHDP were split into a training (63%), validation (27%) and test set (10% of samples). For IHDP we used exactly the same splits as previously used by Shalit et al. (2017). We repeated experiments on IHDP, Jobs, News and TCGA 1000, 10, 50, and 5 times, respectively. We reassigned outcomes and treatments with a new random seed for each repetition.

4.2 Experimental Setup

Models. We evaluated PM, ablations, baselines, and all relevant state-of-the-art methods: kNN (Ho et al. (2007)), BART (Chipman et al. (2010); Chipman & McCulloch (2016)), Random Forests (RF) (Breiman (2001)), CF (Wager & Athey (2017)), GANITE (Yoon et al. (2018)), Balancing Neural Network (BNN) (Johansson et al. (2016)), TARNET (Shalit et al. (2017)), Counterfactual Regression Network using the Wasserstein regulariser (CFRNET$_{\text{Wass}}$) (Shalit et al. (2017)), CEVAE (Louizos et al. (2017)), CMGP (Alaa & van der Schaar (2017)), and PD (Alaa et al. (2017)). We trained a Support Vector Machine (SVM) with probability estimation (Pedregosa et al. (2011)) to estimate $p(t|X)$ for PM on the training set. We also evaluated preprocessing the entire training set with PSM using the "MatchIt" package (Ho et al. (2011)) before training a TARNET model (Appendix E), an ablation of PM where we matched on the covariates X directly (+ on X) instead of on the propensity score, and using PM with a simple multi-layer perceptron (+ MLP) architecture that received the treatment option index t_j as an input instead of using the TARNET architecture.

Architectures. To ensure that differences between methods of learning counterfactual representations for neural networks are not due to differences in architecture, we based the neural network architectures for TARNET, CFRNET$_{\text{Wass}}$, PD and PM on a naïve extension (Appendix F) of the TARNET architecture (Shalit et al. (2017)) to the multiple treatment setting. Specifically, the used TARNET architecture featured a shared MLP that was then split into k individual heads, that were themselves MLPs. We used one head network for each available treatment option. Each head was only trained on the samples that have been treated with the respective treatment option, and the

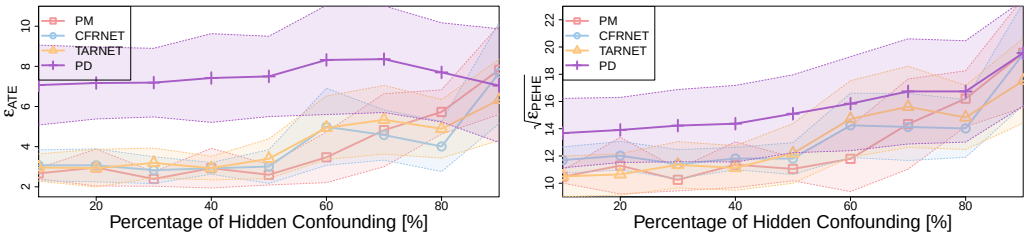


Figure 4: Comparison of several state-of-the-art methods for counterfactual inference on the test set of the TCGA dataset when varying the percentage of hidden confounding (x-axis), i.e. what percentage of the features that were relevant for treatment assignment are visible to the model. Symbols correspond to the mean value of $\hat{\epsilon}_{\text{ATE}}$ (left) and $\sqrt{\hat{\epsilon}_{\text{PEHE}}}$ (right) across 5 repeated runs with new outcomes (lower is better). The shaded area indicates the standard deviation. PM, TARNET and CFRNET are relatively stable up to 60% of hidden confounding. The methods degrade strongly in counterfactual prediction performance at levels of hidden confounding higher than 60%.

Table 3: Comparison of methods for counterfactual inference with two available treatment options on IHDP, Jobs and News-2. We report the mean value $\pm$ the standard deviation of either the $\sqrt{\epsilon_{\text{PEHE}}}$, ϵ_{ATE}, R_{Pol} and ϵ_{ATT} metrics on the test sets over a number of repeated runs. Where available we list the numbers reported by the original authors. n.r. = not reported.

Method	IHDP (1000 repeats)		Jobs (10 repeats)		News-2 (50 repeats)	
	$\sqrt{\epsilon_{\text{PEHE}}}$	ϵ_{ATE}	$R_{\text{Pol}}(\pi)$	ϵ_{ATT}	$\sqrt{\hat{\epsilon}_{\text{PEHE}}}$	$\hat{\epsilon}_{\text{ATE}}$
PM	0.86 ± 0.49	0.24 ± 0.21	0.16 ± 0.09	0.17 ± 0.08	$\mathbf{16.88} \pm 1.12$	$\mathbf{4.15} \pm 0.96$
+ on X	1.38 ± 1.67	0.27 ± 0.35	0.16 ± 0.08	0.19 ± 0.10	17.15 ± 1.24	4.49 ± 1.50
+ MLP	1.86 ± 4.00	0.39 ± 1.00	0.15 ± 0.04	0.17 ± 0.09	18.14 ± 1.50	7.36 ± 2.23
kNN	4.1 ± 0.2	0.79 ± 0.05	0.26 ± 0.0	0.13 ± 0.05	18.14 ± 1.64	7.83 ± 2.55
PSM	2.70 ± 3.85	0.49 ± 0.81	0.20 ± 0.09	0.42 ± 0.22	17.40 ± 1.30	4.89 ± 2.39
RF	6.6 ± 0.3	0.96 ± 0.06	0.28 ± 0.0	0.09 ± 0.04	17.39 ± 1.24	5.50 ± 1.20
CF	3.8 ± 0.2	0.40 ± 0.03	0.20 ± 0.0	0.07 ± 0.03	18.36 ± 1.73	8.48 ± 2.46
BART	2.3 ± 0.1	0.34 ± 0.02	0.25 ± 0.0	0.08 ± 0.03	18.53 ± 2.02	5.40 ± 1.53
GANITE	2.4 ± 0.4	0.49 ± 0.05	$\mathbf{0.14} \pm 0.01$	0.06 ± 0.03	18.28 ± 1.66	4.65 ± 2.12
BNN	2.1 ± 0.1	0.42 ± 0.03	0.24 ± 0.0	0.09 ± 0.04	n.r.	n.r.
PD	n.r.	n.r.	n.r.	n.r.	17.52 ± 1.62	4.69 ± 3.17
TARNET	0.95 ± 0.02	0.28 ± 0.01	0.21 ± 0.0	0.11 ± 0.04	17.17 ± 1.25	4.58 ± 1.29
CFRNET$_{\text{Wass}}$	$\mathbf{0.76} \pm 0.02$	0.27 ± 0.01	0.21 ± 0.0	0.09 ± 0.03	16.93 ± 1.12	4.54 ± 1.48
CEVAE	2.7 ± 0.1	0.46 ± 0.02	0.26 ± 0.0	$\mathbf{0.03} \pm 0.01$	n.r.	n.r.
CMGP	0.77 ± 0.11	$\mathbf{0.13} \pm 0.12$	0.24 ± 0.05	0.09 ± 0.07	n.r.	n.r.

common MLP was trained on all samples. See Shalit et al. (2017) or Appendix F for a more detailed description of the TARNET architecture.

Hyperparameters. For the IHDP, Jobs, News and TCGA dataset we respectively used 30, 30, 10, and 5 optimisation runs for each method using randomly selected hyperparameters from predefined ranges (Appendix G). We selected the best model across the runs based on validation set $\hat{\epsilon}_{\text{NN-PEHE}}$.

Metrics. We calculated the ϵ_{PEHE} and ϵ_{ATE} for datasets for which we know the outcome generating process. For the Jobs dataset, we did not have access to counterfactual outcomes. We therefore calculated the ATT $= \frac{1}{|T_1 \cap E|} \sum_{n=0}^{N} y_1(n) - \frac{1}{|T_0 \cap E|} \sum_{n=0}^{N} y_0(n)$ where E was the randomised subset of the dataset, and reported the error in estimating ATT $\epsilon_{\text{ATT}} = |\text{ATT} - \frac{1}{|T_1 \cap E|} \sum_{n=0}^{N} \hat{y}_1(n)|$, and the policy risk $R_{\text{Pol}}(\pi)$ (Shalit et al. (2017); Yoon et al. (2018)). The policy risk is the average loss in value when treating according to the policy π implied by an ITE estimator (Appendix H).

5 Results and Discussion

Counterfactual Inference. We evaluated the performance for counterfactual inference of the listed models in the binary setting (Table 3) and multiple treatment setting (Table 4). On IHDP, PM reached the third best performance in terms of $\sqrt{\epsilon_{\text{PEHE}}}$ after CFRNET and CMGP, and the second best ϵ_{ATE} after CMGP. On Jobs, PM reached the second best $R_{\text{Pol}}(\pi)$ after GANITE. On the binary News-2, PM outperformed all other methods in terms of $\sqrt{\epsilon_{\text{PEHE}}}$ and ϵ_{ATE}. On the News-4/8/16 datasets with more than two treatments, PM consistently outperformed all other methods - in some cases by a large margin - on both metrics with the exception of the News-4 dataset, where PM came second to PD. The strong performance of PM across a wide range of datasets with varying amounts of treatment options is remarkable considering how simple it is compared to other, highly specialised methods. Notably, PM consistently outperformed both CFRNET, with the exception of the IHDP dataset, which accounted for covariate imbalances between treatments via regularisation rather than matching, and PSM, which accounted for covariate imbalances by preprocessing the training set with a propensity score matching algorithm (Ho et al. (2011)). We also found that matching on the propensity score was consistently better than matching on the covariates (+ on X), and that using PM with the TARNET architecture outperformed the MLP (+ MLP) in almost all cases, with the exception of Jobs. We conclude that matching on the propensity score and using the TARNET architecture is a sensible default configuration, particularly when X is high-dimensional.

Model Selection. To assess whether NN-PEHE is more suitable for model selection for counterfactual inference than MSE, we compared their respective correlations with the PEHE on IHDP. We found that NN-PEHE correlates significantly better with the PEHE than MSE (Figure 1).

Table 4: Comparison of several state-of-the-art methods for counterfactual inference with more than two available treatment options on the News-4, News-8 and News-16 datasets. We report the mean value $\pm$ the standard deviation of either the $\sqrt{\hat{\epsilon}_{\text{mPEHE}}}$ and $\hat{\epsilon}_{\text{mATE}}$ metrics over 50 repeated runs.

Method	News-4 ($\kappa = 10$)		News-8 ($\kappa = 10$)		News-16 ($\kappa = 7$)	
	$\sqrt{\hat{\epsilon}_{\text{mPEHE}}}$	$\hat{\epsilon}_{\text{mATE}}$	$\sqrt{\hat{\epsilon}_{\text{mPEHE}}}$	$\hat{\epsilon}_{\text{mATE}}$	$\sqrt{\hat{\epsilon}_{\text{mPEHE}}}$	$\hat{\epsilon}_{\text{mATE}}$
PM	21.35 ± 2.55	9.88 ± 2.57	$\mathbf{20.89} \pm 1.84$	$\mathbf{6.51} \pm 1.52$	$\mathbf{20.27} \pm 1.66$	$\mathbf{6.17} \pm 1.45$
+ on X	23.00 ± 2.08	13.43 ± 1.88	22.04 ± 2.11	9.19 ± 1.97	20.52 ± 2.13	7.62 ± 1.80
+ MLP	26.35 ± 2.32	17.66 ± 2.49	25.38 ± 2.22	14.33 ± 2.26	26.51 ± 2.17	16.53 ± 1.99
kNN	27.92 ± 2.44	19.40 ± 3.12	26.20 ± 2.18	15.11 ± 2.34	27.64 ± 2.40	17.27 ± 2.10
PSM	37.26 ± 2.28	30.19 ± 2.47	30.50 ± 1.70	22.09 ± 1.98	28.17 ± 2.02	18.81 ± 1.74
RF	26.59 ± 3.02	18.03 ± 3.18	23.77 ± 2.14	12.40 ± 2.29	26.13 ± 2.48	15.91 ± 2.00
CF	27.28 ± 2.59	19.04 ± 3.15	25.76 ± 2.05	15.12 ± 2.20	26.85 ± 2.29	17.03 ± 1.99
BART	26.41 ± 3.10	17.14 ± 3.51	25.78 ± 2.66	14.80 ± 2.56	27.45 ± 2.84	17.50 ± 2.49
GANITE	24.50 ± 2.27	13.84 ± 2.69	23.58 ± 2.48	11.20 ± 2.84	25.12 ± 3.53	13.20 ± 3.28
PD	$\mathbf{20.88} \pm 3.24$	$\mathbf{8.47} \pm 4.51$	21.19 ± 2.29	7.29 ± 2.97	22.28 ± 2.25	10.65 ± 2.22
TARNET	23.40 ± 2.20	13.63 ± 2.18	22.39 ± 2.32	9.38 ± 1.92	21.19 ± 2.01	8.30 ± 1.66
CFRNET$_{\text{Wass}}$	22.65 ± 1.97	12.96 ± 1.69	21.64 ± 1.82	8.79 ± 1.68	20.87 ± 1.46	8.05 ± 1.40

Number of Matches per Minibatch. To assess the impact of matching fewer than 100% of all samples in a batch, we evaluated PM on News-8 trained with varying percentages of matched samples on the range 0 to 100% in steps of 10% (Figure 2). We found that including more matches indeed consistently reduces the counterfactual error up to 100% of samples matched. Interestingly, we found a large improvement over using no matched samples even for relatively small percentages (<40%) of matched samples per batch. This shows that propensity score matching within a batch is indeed highly effective at improving the training of neural networks for counterfactual inference.

Treatment Assignment Bias. To assess how the predictive performance of the different methods is influenced by increasing amounts of treatment assignment bias, we evaluated their performances on News-8 while varying the assignment bias coefficient κ on the range of 5 to 20 (Figure 3). We found that PM handles high amounts of assignment bias better than existing state-of-the-art methods.

Hidden Confounding. To assess the sensitivity of the different methods to hidden confounding, we successively retrained the models on a reduced set of covariates x_i in increasing steps of 10% ($= 2053$ gene loci) in the range of 10% to 90% (Figure 4). We removed the covariates from the model but kept them for computing assignments and outcomes, i.e. they were hidden confounders. The results showed that PM was competitive with TARNET and CFRNET in terms of robustness to hidden confounding. Notably, PM, TARNET and CFRNET were relatively stable up to levels of 60% of hidden confounding, likely because some gene loci are highly correlated.

Limitations. A limitation of this work is that the theory of matching to balance the covariates only holds in the strongly ignorable setting, i.e. under the assumption that there are no unobserved confounders. However, our experiments (Figure 4) and related works showed that even the presence of large amounts of hidden confounders may not necessarily decrease the performance of ITE estimators in practice if we observe proxy variables (Montgomery et al. (2000); Louizos et al. (2017)). In future work, we would like to integrate PM with diagnostic tools that address hidden confounding (Kallus & Zhou (2018)), and the ability to learn using distant supervision (Schwab et al. (2018)).

6 Conclusion

We presented PM, a new and simple method for training neural networks for estimating ITEs from observational data that extends to any number of available treatment options. In addition, we extended the PEHE metric to settings with more than two treatments, and introduced a nearest neighbour approximation of PEHE and mPEHE that can be used for model selection without having access to counterfactual outcomes. We performed experiments on several real-world and semi-synthetic datasets that showed that PM outperforms a number of more complex state-of-the-art methods in inferring counterfactual outcomes. We also found that the NN-PEHE correlates significantly better with real PEHE than MSE, that including more matched samples in each minibatch improves the learning of counterfactual representations, and that PM handles an increasing treatment assignment bias better than existing state-of-the-art methods. PM may be used for settings with any amount of treatment options, is compatible with any existing neural network architecture, simple to implement, and does not introduce any additional hyperparameters or computational complexity.

REFERENCES

Ahmed M Alaa and Mihaela van der Schaar. Bayesian inference of individualized treatment effects using multi-task gaussian processes. In *Advances in Neural Information Processing Systems*, pp. 3424–3432, 2017.

Ahmed M Alaa, Michael Weisz, and Mihaela van der Schaar. Deep counterfactual networks with propensity-dropout. *arXiv preprint arXiv:1706.05966*, 2017.

Susan Athey, Julie Tibshirani, and Stefan Wager. Generalized random forests. *arXiv preprint arXiv:1610.01271*, 2016.

Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

Hugh Chipman and Robert McCulloch. BayesTree: Bayesian additive regression trees. *R package version 0.3-1.4*, 2016.

Hugh A Chipman, Edward I George, Robert E McCulloch, et al. BART: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

Dominik Csiba and Peter Richtárik. Importance sampling for minibatches. *Journal of Machine Learning Research*, 19(27), 2018.

Rajeev H Dehejia and Sadek Wahba. Propensity score-matching methods for nonexperimental causal studies. *Review of Economics and statistics*, 84(1):151–161, 2002.

Vincent Dorie. NPCI: Non-parametrics for causal inference, 2016. URL `https://github.com/vdorie/npci`.

Michele Jonsson Funk, Daniel Westreich, Chris Wiesen, Til Strmer, M. Alan Brookhart, and Marie Davidian. Doubly robust estimation of causal effects. *American Journal of Epidemiology*, 173 (7):761–767, 2011.

Jennifer L Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

Daniel E Ho, Kosuke Imai, Gary King, and Elizabeth A Stuart. Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference. *Political analysis*, 15(3): 199–236, 2007.

Daniel E Ho, Kosuke Imai, Gary King, Elizabeth A Stuart, et al. MatchIt: nonparametric preprocessing for parametric causal inference. *Journal of Statistical Software*, 42(8):1–28, 2011.

Fredrik Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning*, pp. 3020–3029, 2016.

Nathan Kallus. Recursive partitioning for personalization using observational data. In *International Conference on Machine Learning*, 2017.

Nathan Kallus and Angela Zhou. Confounding-robust policy improvement. *Advances in Neural Information Processing Systems*, 2018.

Adam Kapelner and Justin Bleich. bartMachine: Machine learning with Bayesian additive regression trees. *arXiv preprint arXiv:1312.2171*, 2013.

Robert J LaLonde. Evaluating the econometric evaluations of training programs with experimental data. *The American economic review*, pp. 604–620, 1986.

Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. In *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

Mark R Montgomery, Michele Gragnolati, Kathleen A Burke, and Edmundo Paredes. Measuring living standards with proxy variables. *Demography*, 37(2):155–174, 2000.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Paul R. Rosenbaum and Donald B. Rubin. The central role of the propensity score in observational studies for causal effects. *Biometrika*, 70(1):41–55, 1983.

Donald B Rubin. Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331, 2005.

Patrick Schwab, Emanuela Keller, Carl Muroi, David J Mack, Christian Strässle, and Walter Karlen. Not to Cry Wolf: Distantly Supervised Multitask Learning in Critical Care. In *International Conference on Machine Learning*, 2018.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *International Conference on Machine Learning*, 2017.

Jeffrey A Smith and Petra E Todd. Does matching overcome LaLonde's critique of nonexperimental estimators? *Journal of Econometrics*, 125(1-2):305–353, 2005.

Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 2017.

John N Weinstein, Eric A Collisson, Gordon B Mills, Kenna R Mills Shaw, Brad A Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, Joshua M Stuart, Cancer Genome Atlas Research Network, et al. The cancer genome atlas pan-cancer analysis project. *Nature genetics*, 45(10): 1113–1120, 2013.

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

SUPPLEMENTARY MATERIAL FOR: "PERFECT MATCH: A SIMPLE METHOD FOR LEARNING REPRESENTATIONS FOR COUNTERFACTUAL INFERENCE WITH NEURAL NETWORKS"

Anonymous authors
Paper under double-blind review

A PRECISION IN ESTIMATION OF HETEROGENOUS EFFECT (PEHE)

When the underlying noiseless distributions μ_j are not given, PEHE is calculated using:

$$\hat{\epsilon}_{\text{PEHE}} = \frac{1}{N}\sum_{n=0}^{N}\Big([y_1(n) - y_0(n)] - [\hat{y}_1(n) - \hat{y}_0(n)]\Big)^2 \tag{A.1}$$

B AVERAGE TREATMENT EFFECT (ATE)

The ATE measures the average difference in effect across the whole population (Hill (2011)). It can be useful indicator of how well an ITE estimator performs at comparing two treatments across the entire population.

$$\epsilon_{\text{ATE}} = ||\frac{1}{N}\sum_{n=0}^{N}\mathbb{E}_{y(n)\sim\mu(n)}[y(n)] - \frac{1}{N}\sum_{n=0}^{N}\hat{y}(n)||_2^2 \tag{B.1}$$

Similar to Equation (A.1), the counterfactual treatment effect can be estimated using the noisy ground truth outcomes if the underlying noiseless distributions μ_j are not given:

$$\hat{\epsilon}_{\text{ATE}} = ||\frac{1}{N}\sum_{n=0}^{N}y(n) - \frac{1}{N}\sum_{n=0}^{N}\hat{y}(n)||_2^2 \tag{B.2}$$

C PSEUDOCODE PERFECT MATCH (PM)

Algorithm S1 outlines the procedure of using PM to augment a batch with propensity score matches for training a neural network with stochastic gradient descent. We trained a Support Vector Machine (SVM) with probability estimation as propensity score estimator E_{PS}. To speed up recalling nearest neighbours, we additionally prepared an index per treatment into X_{train} sorted by propensity score. At augmentation time, we used binary search on this index to find the nearest neighbours by propensity score in O(log N_t) where N_t is the number of samples X in X_{train} assigned to the treatment group t. To avoid overfitting to specific edge samples when propensity scores are not distributed evenly in the training set, we chose at random from one of the 6 closest samples by propensity score. For the "+ on X" model we matched on the covariates X using the Mahalanobis distance for IHDP and Jobs, and the Euclidean distance for News. We used the Euclidean distance on News because the covariance matrix for the word counts vectors X on the News datasets was singular.

Algorithm S1 Batch Augmentation using Perfect Match (PM). After augmentation, each batch contains an equal number of samples from each treatment group and the covariates x_i across all treatment groups are approximately balanced.

Input: Batch of B random samples X_{batch} with assigned treatments t, training set X_{train} of N samples, Number of treatment options k, propensity score estimator E_{PS} to calculate the probability $p(t|X)$ of treatment assigned given a sample X
Output: Batch X_{out} consisting of $B \times k$ matched samples

```
procedure PERFECT_MATCH:
    X_out ← Empty
    for sample X with treatment t in X_batch do
        p(t|X) ← E_PS(X)
        for i = 0 to k − 1 do
            if i ≠ t then
                ps_i ← p(t|X)_i
                X_matched ← get closest match to propensity score ps_i with treatment i from X_train
                Add sample X_matched to X_out
        Add X to X_out
```

D NEAREST NEIGHBOUR APPROXIMATION OF PEHE FOR MULTIPLE TREATMENTS (NN-MPEHE)

The $\hat{\epsilon}_{\text{NN-PEHE}}$ metric can be extended to the multiple treatment setting by considering the mean $\hat{\epsilon}_{\text{NN-PEHE}}$ between all $\binom{k}{2}$ possible pairs of treatments:

$$\hat{\epsilon}_{\text{NN-mPEHE}} = \frac{1}{\binom{k}{2}} \sum_{i=0}^{k-1} \sum_{j=0}^{i-1} \hat{\epsilon}_{\text{NN-PEHE},i,j} \tag{D.1}$$

E PSEUDOCODE PROPENSITY SCORE MATCHING (PSM)

Algorithm S2 outlines the procedure of preprocessing a training set using PSM. We used the MatchIt package (Ho et al. (2011)) on setting "nearest" to find matches between sets of samples from two treatment groups (function `get_matched_samples`).

Algorithm S2 Preprocessing a training set using Propensity Score Matching (PSM). After preprocessing, the training set contains an equal number of samples from each treatment group and the covariates x_i across all treatment groups are approximately balanced.

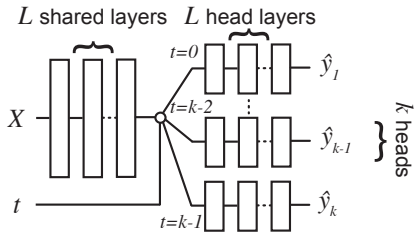
Input: N training samples X_{train} with assigned treatments t, Number of treatment options k, function `get_matched_samples` to find nearest matches by propensity score between sets of samples
Output: Preprocessed training set X_{out} consisting of matched samples from each treatment group

```
procedure PREPROCESS_TRAINING_SET:
    t_base ← index of treatment with smallest number of samples in X_train
    X_base ← all samples in X_train with treatment t_base
    for i = 0 to k − 1 do
        if i ≠ t_base then
            X_current ← all samples in X_train with treatment i
            X_matched ← get_matched_samples(X_base, X_current)
            Add all samples in X_matched to X_out
    Add all samples in X_base to X_out
```

F TARNET, CFRNET AND PD FOR MULTIPLE TREATMENTS

We naïvely extended the TARNET architecture to the multiple treatment setting (Figure S1) because the original TARNET (Shalit et al. (2017)) was limited to the binary treatment setting. To extend TARNET to the multiple treatment setting, we used k head networks, one for each treatment option. We used ELU nonlinearities between the L hidden layers with M hidden units. L and M were hyperparameters that we optimised during hyperparameter optimisation (Section G). We did not use batch normalisation (BN). To extend CFRNET to the multiple treatment setting, we defined the first treatment option as the control treatment and regularised all treatment options to have the same activation distribution in the topmost shared layer (Shalit et al. (2017)). For PD, we only used propensity dropout and the propensity estimation network. We did not make use of the alternating training schedule proposed in (Alaa et al. (2017)). The "+ MLP" model was a simple MLP with L hidden layers of M hidden units that received the treatment option index t_j as an additional input along with the covariates X, and output a k-dimensional potential outcome vector $\hat{Y}$. The MLP used ELU nonlinearities between the L hidden layers, and also did not use BN.

Figure S1: The TARNET architecture with k heads for multiple treatments.



G HYPERPARAMETERS

To ensure a fair comparison, we used a standardised approach to hyperparameter optimisation for those methods for which we did not have previously reported performance numbers. In particular, each method we trained was optimised over the same amount of hyperparameter optimisation runs. For the methods that used neural network models (TARNET, CFRNET, PD, PSM, including "+ on X" and "+ MLP", and PM), we chose hyperparameters at random from predefined ranges (Table S1). For CFRNET, we additionally varied the weight of the imbalance penalty at random between 0.1, 1.0, and 10.0. All methods that used a neural network model used the TARNET architecture to ensure differences in performance are not due to architectural differences. To train PM for the Jobs dataset, we used fixed hyperparameters: A batch size of 50, 60 hidden units per layer, and 3 hidden layers. Note that on IHDP and Jobs we trained only PM and PSM, all other numbers were taken from previous reports by the respective original authors. For optimisation, we used the Adam optimiser with a learning rate of 0.001 for a maximum of 100 (News, TCGA) or 400 (IHDP, Jobs) with an early stopping patience of 30 on the factual MSE. We used the default hyperparameters for BART and CF from the "bartMachine" (Kapelner & Bleich (2013)) and "grf" (Athey et al. (2016)) R-packages. For GANITE, we used our own implementation since there was no open source implementation available, and - in addition to the parameters in Table S1 - optimised over the supervised loss weights α and β (Yoon et al. (2018)) between 0.1, 1, and 10. For the GANITE generators and discriminators, we used MLP architectures with L hidden layers of M hidden units each.

Table S1: Hyperparameter ranges used in the performed experiments.

Hyperparameter	IHDP	News/TCGA
Batch size B	4, 8, 50, 100	50
Number of units per hidden layer M	50, 100, 200	40, 60, 80
Number of hidden layers L	1, 2, 3	2, 3

H POLICY RISK $R_{\text{POL}}(\pi)$

We define the policy risk $R_{\text{Pol}}(\pi)$ as the expected loss in value when treating according to the policy π implied by an ITE estimator (Shalit et al. (2017)):

$$R_{\text{Pol}}(\pi) = 1 - (\mathbb{E}[y_1|\pi(X) = 1] \cdot p(\pi = 1) + \mathbb{E}[y_0|\pi(X) = 0] \cdot p(\pi = 0)) \tag{H.1}$$

where the policy $\pi(X)$ for a sample X implied by an ITE estimator that produces potential outcomes $\hat{y}$ is defined as:

$$\pi(X) = \begin{cases} 1 & \text{if } \hat{y}_{1,X} - \hat{y}_{0,X} > \lambda \quad \text{(to treat)} \\ 0 & \text{otherwise} \quad \text{(to not treat)} \end{cases} \tag{H.2}$$