# PairNet: Training with Observed Pairs to Estimate Individual Treatment Effect

**Lokesh Nagalapatti** [*][1] **Pranava Singhal** [*][1] **Avishek Ghosh** [1] **Sunita Sarawagi** [1]

## Abstract

Given a dataset of individuals each described by a covariate vector, a treatment, and an observed outcome on the treatment, the goal of the individual treatment effect (ITE) estimation task is to predict outcome changes resulting from a change in treatment. A fundamental challenge is that in the observational data, a covariate's outcome is observed only under one treatment, whereas we need to infer the difference in outcomes under two different treatments. Several existing approaches address this issue through training with inferred pseudo-outcomes, but their success relies on the quality of these pseudo-outcomes. We propose PairNet, a novel ITE estimation training strategy that minimizes losses over pairs of examples based on their factual observed outcomes. Theoretical analysis for binary treatments reveals that PairNet is a consistent estimator of ITE risk, and achieves smaller generalization error than baseline models. Empirical comparison with thirteen existing methods across eight benchmarks, covering both discrete and continuous treatments, shows that PairNet achieves significantly lower ITE error compared to the baselines. Also, it is model-agnostic and easy to implement. We release the code at the URL: https://github.com/nlokeshiisc/pairnet_release.

## 1. Introduction

Many applications in medicine, finance, and retail require the ability to predict the change in outcome resulting from changing actions or treatments. Classical methods relied on expensive randomized control trial experiments, but with the increasing availability of large observational datasets, a recent research focus is harnessing this observational data to train models for estimating treatment effects. For example, a retail store might have millions of customers whose click-through rates on advertisements have been observed under various settings of treatments such as discounts. Using such datasets it may be possible to train a model to estimate the effect of discounts on click-through rate. However, a challenge is that an individual is observed only under one treatment, and there is no direct supervision of how an individual's outcome could change with changing treatment. Thus, one baseline approach (Shalit et al., 2017; Nie et al., 2021; Shalit et al., 2016; Chauhan et al., 2023; Curth & van der Schaar, 2021; Zhang et al., 2020; Shi et al., 2019) is to train a model to fit outcomes separately for each treatment, and during inference output the difference of predicted outcomes under the two treatments. Another major line of work is to impute pseudo-outcomes for missing treatments in the training dataset and directly supervise outcome differences using them. Imputation methods include meta-learners (Künzel et al., 2019; Nie & Wager, 2021; Curth & van der Schaar, 2023), matching methods (Stuart, 2010; Rosenbaum & Rubin, 1983; Iacus et al., 2012; Schwab et al., 2018; Kallus, 2020), and generative models such as GANs (Bica et al., 2020; Louizos et al., 2017). However, the success of such methods depends on the quality of the inferred pseudo-outcomes.

We propose an alternative training strategy called PairNet, that avoids committing to noisy supervision from pseudo-outcomes and instead works only with observed factual outcomes. Unlike all previous methods, PairNet imposes a loss on the *difference of outcomes of pairs of instances*. The pairs are chosen to be close in the covariate space while having different treatments. We show that such paired instance-based training more closely aligns with the ITE estimation task than existing methods.

We theoretically analyze Pair loss for binary treatments and bound the ITE risk. These bounds, expressed in the form of IPM distance between the provided training distribution and a neighborhood distribution, are shown to be tighter when compared to the baseline factual model, which relies on the IPM distance between treated and control distributions. Additionally, we establish that PairNet serves as a consistent estimator of treatment effect.

In summary, we make the following contributions:

[*]Equal contribution [1]IIT Bombay. Correspondence to: Lokesh Nagalapatti <nlokeshiisc@gmail.com>, Pranava Singhal <pranava.psinghal@gmail.com>, Avishek Ghosh <avishek_ghosh@iitb.ac.in>, Sunita Sarawagi <sunita@iitb.ac.in>.

*Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

1. We introduce PairNet, a novel approach that *only* applies factual losses to observed instance pairs in contrast to several existing methods that infer pseudo-outcomes. PairNet is *model-agnostic* and applies to *both* discrete and continuous treatments.
2. We theoretically show that the ITE risk can be upper bounded by Pair loss and the distributional distance of near neighbours with contrasting treatments. We further show that PairNet is a consistent estimator of ITE under commonly used strict overlap assumption and offers tighter generalization bounds that factual loss.
3. We compare PairNet with eleven prior methods on three benchmarks for binary treatments and two prior methods on five benchmarks for continuous treatments. We observe that PairNet provides significant gains. We conduct several experiments to explain the superior performance of PairNet and analyze its sensitivity.

## 2. Problem Statement

We follow the Neyman-Rubin potential outcomes framework (Rubin, 2005) where an individual with observed covariates $\mathbf{x}\in\mathcal{X}$, when subjected to a treatment $t\in\mathcal{T}$, exhibits an outcome $Y(t)\in\mathbb{R}$. The space of treatments $\mathcal{T}$ could be binary ($\mathcal{T}=\{0,1\}$) or continuous ($\mathcal{T}=[0,1]$). We are given an observational dataset $D=\{(\mathbf{x}_i,t_i,y_i):i\in 1...N\}$ where $(\mathbf{x},t)$ are samples drawn from a joint distribution $P(X,T)$ such that $X \not\perp\!\!\!\perp T$, and $y_i$ is the observed outcome. We denote the covariate distribution $P(X=\mathbf{x})$ as $p(\mathbf{x})$, and the conditional distribution $P(X=\mathbf{x}|T=t)$ as $p_t(\mathbf{x})$. The marginal treatment distribution is $P(T=t)=u_t$. Let the expected outcome when a covariate $\mathbf{x}$ is given a treatment $t$ be denoted as $\mu^*(\mathbf{x},t)=\mathbb{E}[(Y(t)|\mathbf{x})]$. Our goal is to learn a model $\hat{\tau}(\mathbf{x},t,t')$ that estimates the change in outcome $\tau^*(\mathbf{x},t,t')=\mu^*(\mathbf{x},t)-\mu^*(\mathbf{x},t')$ when a test $(\mathbf{x},t)\sim P(X,T)$ is given a new treatment $t'$ that is sampled arbitrarily. Solving this problem requires us to minimize the following **ITE risk**:

$$\mathbb{E}_{P(X,T),T'\neq T}[\|(\tau^*(X,T,T')-\hat{\tau}(X,T,T'))\|_2] \quad (1)$$

**Assumptions.** Like in prior work we make the following assumptions for identifying ITE from an observational dataset: A1 *Overlap of treatment:* Every individual has a non-zero probability of being assigned any treatment, i.e., $P(t|\mathbf{x})\in(0,1);\forall\mathbf{x}\in\mathcal{X},t\in\mathcal{T}$. A2 *Consistency:* The observed outcome is the same as the potential outcome, i.e., when an individual is given a treatment $t$, we observe $Y(t)$. A3 *Unconfoundedness:* The observed covariates $X$ block all backdoor paths between the treatments $T$ and outcomes $Y$.

**Challenges.** The fundamental challenge in estimating $\hat{\tau}$ from the observational dataset is that for a given $\mathbf{x}$, we observe its outcome under only one treatment sampled according to $P(t|\mathbf{x})=\pi_t(\mathbf{x})$. Thus, we cannot directly supervise $\hat{\tau}(\mathbf{x},t,t')$ using $D$. A common practice is to estimate $\hat{\mu}(\mathbf{x},t)$ instead by minimizing the following **factual** loss:

$$\hat{\mu}=\underset{\mu}{\operatorname{argmin}}\sum_{i=1}^{N}(y_i-\mu(\mathbf{x}_i,t_i))^2 \quad (2)$$

and then estimate the treatment effect as

$$\hat{\tau}(\mathbf{x},t,t')=\hat{\mu}(\mathbf{x},t)-\hat{\mu}(\mathbf{x},t') \quad (3)$$

We call this the **Factual model**. The main limitation of the factual loss-based training is that the loss is independent for each treatment whereas, for ITE risk, we wish to minimize error in the difference in outcomes of two treatments $t,t'$. The gap between factual risk and ITE risk gets even worse because of confounding ($T \not\perp\!\!\!\perp X$) causing $p_t(\mathbf{x})\neq p_{t'}(\mathbf{x})$ in general. For example, for binary treatment, the $\hat{\mu}(\mathbf{x},1)$ trained on examples sampled from $p_1(\mathbf{x})$, may perform poorly when during inference it is deployed for treatment effect estimation on $\mathbf{x}\sim p_0(\mathbf{x})$.

## 3. Related Work

We categorize the extensive literature on ITE estimation based on whether they train with pseudo-outcomes, or not.

### 3.1. Training with pseudo-outcomes

A common strategy to address the ITE task is to estimate the pseudo-outcomes for missing treatments and then use them to minimize the ITE risk in Eq. 1. These methods differ primarily in the method in which the pseudo-outcomes are inferred.

**Meta Learners.** One class of approaches directly models the treatment effect $\hat{\tau}$ under the assumption that the treatment effect function $\tau$ is simpler than the individual potential outcome functions $\mu_t$ (Gao & Han, 2020; Curth & van der Schaar, 2021). A popular strategy is using two-stage meta-learning. First, train a Factual model $\hat{\mu}(\mathbf{x},t)$ to estimate outcome using $D$ as shown in Eq 2, and then train the ITE model $\hat{\tau}(\mathbf{x},t,t')$ using pseudo ITE $y_i(t_i)-\hat{\mu}(\mathbf{x}_i,t')$. X-Learner (Künzel et al., 2019) is one of the earliest such meta-learners. DR-Learner (Kennedy, 2020) is similar but adds a propensity score model in the first stage to synthesize doubly robust ITE labels $\hat{\tau}$. R-Learner(Nie & Wager, 2021) employs Robinson's error decomposition to directly model $\tau$.

**Matching Methods.** Matching methods (Stuart, 2010; Rosenbaum & Rubin, 1983; Iacus et al., 2012; Schwab et al., 2018; Kallus, 2020) impose losses on missing treatments by borrowing outcomes from near neighbors in the dataset. They use techniques like propensity scores, exact matching, stratification, covariate distance metrics, etc., to form neighbors across treatment groups. However, if the distance metric connecting neighbors is flawed, the inferred outcomes can be wrong leading to faulty supervision. While PairNet also creates neighbors, it imposes losses solely using the observed outcomes in the dataset, and thus the accuracy of PairNet is unaffected by the correctness of any pseudo outcomes.

**Generative methods.** GANITE (Yoon et al., 2018) uses GANs to synthesize pseudo-outcomes and uses them to learn $\tau$. SciGAN (Bica et al., 2020) extends GANITE (Yoon et al., 2018) for continuous treatments. Gaussian Processes are employed in (Alaa & Van Der Schaar, 2017; Alaa & van der Schaar, 2017; Zhang et al., 2020) and Variational Autoencoders (VAEs) are explored in (Louizos et al., 2017; Rissanen & Marttinen, 2021; Lu et al., 2020).

### 3.2. Training without Pseudo-outcomes

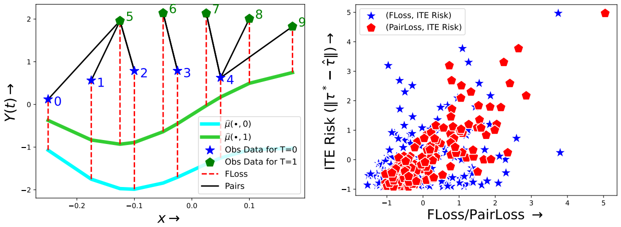
These methods address the treatment confounding challenge by learning shared representations with regularizers to correct the effect of confounding. A popular method is TARNet (Shalit et al., 2016) which uses a shared representation layer and defines separate prediction heads for each treatment group. CFRNet (Shalit et al., 2017) additionally applies IPM regularizers to balance representations. Yao et al. (2018) encourages representation similarity based on propensity scores. FlexTENet (Curth & van der Schaar, 2021) introduces further sharing of parameters across treatment-specific layers. DragonNet (Shi et al., 2019) introduces Targeted Regularizers for CATE estimation. **Weighting methods:** Certain methods account for confounding by imposing weighted factual losses (Hassanpour & Greiner, 2019a;b; Jung et al., 2020; Ozery-Flato et al., 2018; Shi et al., 2019). DragonNet (Shi et al., 2019) is a doubly robust method that implements the Augmented Inverse Probability Weighted estimator. But these methods depend on propensity scores which are often not well-calibrated.

**Continuous Treatments Effects.** DRNet (Schwab et al., 2020) first handled continuous treatments using a TARNet-like architecture with binned treatments. VCNet uses spline basis expansion for $t$, with regularizers as either Targeted(Nie et al., 2021) or the Hilbert Schmidt Independence Criterion (HSIC) (Bellot et al., 2022). GIKS (Nagalapatti et al., 2024) uses data augmentation to break the confounding in the training dataset. Zhang et al. (2022) proposes a transformer-based model designed specifically for text data.

## 4. The Pair Loss

Unlike existing approaches, that attempt to impute outcomes of missing treatment $t'$, PairNet minimizes the difference in observed outcomes of two individuals. The pairs are chosen to be close in the covariate space and are observed under different treatments. Specifically, they are created as follows: For each instance $(\mathbf{x}_i, t_i, y_i) \in D$, sample an alternative treatment $t'$, find instance subsets from $D$ to form neighbor set $\text{Nbr}(\mathbf{x}_i, t') = \{(\mathbf{x}_j, t_j)\}$ such that $t_j \approx t'$ and distance $d(\mathbf{x}_i, \mathbf{x}_j) = ||\psi(\mathbf{x}_i) - \psi(\mathbf{x}_j)||$ is small. Here, $\psi$ denotes any embedding function that can be used to compute distances. This sampling procedure induces a distribution over the neighbours $(\mathbf{x}', t')$ for observed $(\mathbf{x}, t)$ which we denote as $q_t(\mathbf{x}', t'|\mathbf{x}) \propto e^{-d(\mathbf{x},\mathbf{x}')} p_{t'}(\mathbf{x}')$. PairNet models potential outcomes as $\hat{\mu}(\mathbf{x}, t) = \mu(\phi(\mathbf{x}), t)$, similar to representation learning methods where $\phi$ is a representation extraction function shared across treatments, and $\mu$ is the treatment-specific outcome prediction function. It imposes a loss on a pair of instances as follows:

$$\hat{\mu}(\mathbf{x}, t) = \operatorname*{argmin}_{\mu,\phi} \sum_{i=1}^{N} \sum_{t' \neq t_i} \sum_{(\mathbf{x}'_i, t'_i) \in \text{Nbr}(\mathbf{x}_i, t')} ((y_i - y'_i) - (\mu(\phi(\mathbf{x}_i), t_i) - \mu(\phi(\mathbf{x}'_i), t'_i)))^2 \tag{4}$$



(a) Ground truth vs. predicted $\hat{\mu}$ functions

(b) Correlation of Factual & Pair loss with the ITE risk

Figure 1: Motivating Experiment: Panel (a) presents observational data and predicted $\hat{\mu}$ functions, with pairs selected by PairNet indicated by black lines. In Panel (b), we visualize two empirical losses alongside the corresponding ITE risk. We observed a correlation of 0.32 between factual loss and ITE risk, while PairNet achieved a substantially stronger correlation of 0.82. Remarkably, the correlation dropped to 0.45 when Pair loss lacked the residual alignment term, highlighting its importance.

We illustrate the working of the Pair loss in Figure 1a where we assume univariate covariates $\mathbf{x} \in \mathbb{R}$ (shown in black). Suppose we consider the fourth example ($i = 4$) that has $i = 7$ in its neighborhood. The loss for this pair is computed as $((y_4 - y_7) - (\hat{\mu}(x_4, 0) - \hat{\mu}(x_7, 1)))^2$. The pseudo-code for forming pairs and PairNet training is shown in Algorithm 1. We defer the explanation of the algorithm to Appendix A. We infer the treatment effect similar to the Factual model using Equation 3. However, one crucial difference lies in the method of training $\hat{\mu}$. Our loss aligns with the ITE risk defined in Eq. 1, where the only mismatch is that we impose the loss on two different covariates $\mathbf{x}, \mathbf{x}'$ during training. Whereas, during inference, we invoke it on the same $\mathbf{x}$. We get further insights by expanding the Pair loss:

$$\text{Pair Loss}(i, i') = \underbrace{(y_i - \hat{\mu}(\mathbf{x}_i, t_i))^2}_{\text{Factual Loss}(i)} + \underbrace{(y'_i - \hat{\mu}(\mathbf{x}'_i, t'_i))^2}_{\text{Factual Loss}(i')} \underbrace{-2(y_i - \hat{\mu}(\mathbf{x}_i, t_i))(y'_i - \hat{\mu}(\mathbf{x}'_i, t'_i))}_{\text{Residual Alignment}(i, i')} \tag{5}$$

We see that the Pair loss is decomposed into a sum of two factual losses, one on the observed samples and the other on their matched pairs, and the last term acts on error residuals. The last term promotes a positive correlation among error residuals for near covariates which is a necessary inductive bias for ITE estimation. We will analyze the role of this term more formally in Section 5.

**Algorithm 1** PairNet Algorithm

```
Require: Data $D$: $\{(\mathbf{x}_i, t_i, y_i)\}$, distance threshold $\delta_{\text{pair}}$,
    number of pairs $\text{num}_{z'}$, Epochs $E$, $\psi$ for forming pairs
Let $\phi \leftarrow$ rep. network and $\{\mu_t\} \leftarrow$ prediction heads
$D_{\text{trn}}, D_{\text{val}} \leftarrow$ SPLIT($D$, $pc = 0.3$, stratify=$T$)
$D_{\text{val}} \leftarrow$ CREATEPAIRDS($D_{\text{val}}, D, \delta_{\text{pair}}, \text{num}_{z'}, \psi$)
for e $\in [E]$ do
    $D^e_{\text{trn}} \leftarrow$ CREATEPAIRDS($D_{\text{trn}}, D_{\text{trn}}, \delta_{\text{pair}}, \text{num}_{z'}, \psi$)
    for each batch $\{(\mathbf{x}, t, y, \mathbf{x}', t', y')\} \subset D^e_{\text{Trn}}$ do
        $z, z' \leftarrow \phi(\mathbf{x}), \phi(\mathbf{x}')$
        $\hat{y}, \hat{y'} \leftarrow \mu_t(z), \mu_{t'}(z')$
        loss $\leftarrow \mathcal{L}\Big((y - y'), (\hat{y} - \hat{y'})\Big)$
        $\phi, \{\mu_t\} \leftarrow$ GRADDESC(loss)
    end for
    Break if EARLYSTOPPING($D_{\text{val}}, \phi, \{\mu_t\}$)
end for
Return $\phi, \{\mu_t\}$
```

```
function CREATEPAIRDS($D', D, \delta_{\text{pair}}$, $\text{num}_{z'}, \psi$)
$N \leftarrow |D|, D_{\text{pair}} \leftarrow \{\}$
for $(\mathbf{x}'_i, t'_i, y'_i) \in D'$ do
    $d_i[j] \leftarrow$ distance $d(\psi(\mathbf{x}'_i), \psi(\mathbf{x}_j))\ \forall j \in [N]$
    $d_i[j] \leftarrow \infty$ if $t'_i = t_j$
    $q_{t_i}(\mathbf{x}_j|\mathbf{x}_i) \leftarrow$ softmax$(-d_i)$
    $\text{Nbrs}_i \leftarrow$ SAMPLE($q_{t_i}, \text{num}_{z'}$)
    $D_{\text{pair}} \leftarrow D_{\text{pair}} \cup \{(x'_i, t'_i, y'_i, x_j, t_j, y_j)\} \forall j \in \text{Nbrs}_i$
end for
Return $D_{\text{pair}}$ after dropping largest $\delta_{\text{pair}}$ distances.
```

**A simple illustration to show alignment of Pair loss with ITE.** We present a simple study to illustrate that Pair loss is better aligned with ITE risk than Factual loss. Consider binary treatments. Let $\mu^*(\mathbf{x},0) \sim GP(0,\mathcal{K}_\gamma)$ be a smooth function sampled from a Gaussian process with an RBF kernel of width $\gamma$. Next, to capture a common inductive bias (Curth & van der Schaar, 2021; Künzel et al., 2019) that the ITE function $\tau^*$ is simpler than each of the outcome functions, we model $\tau^*(\mathbf{x}) \sim GP(0,\mathcal{K}_\eta)$ where $\eta < \gamma$. This defines $\mu^*(\mathbf{x},1) = \mu^*(\mathbf{x},0) + \tau^*(\mathbf{x})$. We create observation data $D$ by sampling $n_0$ instances from $\mathcal{N}(u,1)$ with $t = 0$, and $n_1$ instances from $\mathcal{N}(u+s,1)$ with $t = 1$. The value $s$ enables us to introduce confounding between the $t$ and $\mathbf{x}$ values. Figure 1a shows an example. We then sample functions from $GP(0,\mathcal{K}_{\gamma'})$ and $GP(0,\mathcal{K}_{\eta'})$ to serve as $\hat{\mu}(\mathbf{x},0)$ and $\hat{\tau}(\mathbf{x})$ respectively. For each sampled function, we measure the actual ITE risk (Eq 1), the Factual loss using $D$, and Pair loss using $D$. Figure 1b shows that Pair loss has a significantly higher agreement with the true ITE risk than the Factual loss. Thus, even without getting into specific network architectures and estimation methods, we observe that as a loss function, training with pairs is better suited for reducing ITE estimation errors.

***Remark**: Our proposal, Pair loss is model-agnostic, and hence we refer to any model that applies Pair loss to train for ITE estimation as PairNet.*

## 5. ITE Risk Bounds for Binary Treatment

We theoretically bound the ITE risk with PairNet's loss and compare it with the Factual model on binary treatments.

**Notation.** Recall that covariates for a treatment $t$ are sampled from $p_t(\mathbf{x}) = p(\mathbf{x}|t)$ and $u_t = p(T = t)$ denotes treatment marginals. True outcome functions are $\mu^*_t(\mathbf{x})$ and estimated functions are $\hat{\mu}_t(\mathbf{x}) = \mu(\phi(\mathbf{x}),t)$. For ease of notation, we assume that $\phi(\mathbf{x})$ is identity, but in the Appendix (Section B) we show how to extend the analysis to $\phi(\mathbf{x})$ under assumptions of (Shalit et al., 2017).

**Definition 5.1.** The *error residual* for an instance $\mathbf{x}$ under a treatment $t$ is defined as $r_t(\mathbf{x}) = \hat{\mu}_t(\mathbf{x}) - \mu^*_t(\mathbf{x})$.

**Definition 5.2.** The *factual errors* under a treatment $t$ are:

$$\epsilon^t_{\text{F}} = \int_{\mathbf{x}} r_t(\mathbf{x})^2 p_t(\mathbf{x}) d\mathbf{x}, \ \ \epsilon_F = \sum_t u_t \epsilon^t_F,$$

**Definition 5.3.** The *ITE risk* is defined in terms of residuals as:

$$\epsilon_{\text{ITE}} = \int_{\mathbf{x}} (r_1(\mathbf{x}) - r_0(\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x}$$

**Definition 5.4.** For an observed instance $\mathbf{x},t$, the induced *neighborhood distribution* from which pairs $\mathbf{x}'$ with $t' = 1-t$ are sampled is represented as $q_t(\mathbf{x}'|\mathbf{x})$. Marginalising over $\mathbf{x}$, we obtain $q_t(\mathbf{x}') = \int_{\mathbf{x}} q_t(\mathbf{x}'|\mathbf{x}) p_t(\mathbf{x}) d\mathbf{x}$.

**Definition 5.5.** PairNet's objective in Eq. 4 can be written as:

$$\epsilon_{\text{pair}} = \sum_t u_t \int_{\mathbf{x}} \int_{\mathbf{x}'} [(r_t(\mathbf{x}) - r_{(1-t)}(\mathbf{x}'))^2] dq_t(\mathbf{x}'|\mathbf{x}) dp_t(\mathbf{x})$$

**Lemma 5.6.** *The difference between ITE Risk and PairNet loss can be expressed as*

$$\epsilon_{ITE} - \epsilon_{pair} = \sum_t u_{(1-t)} \int_{\mathbf{x}} r_t(\mathbf{x})^2 (p_t(\mathbf{x}) - q_t(\mathbf{x})) d\mathbf{x}$$
$$+ \sum_t 2u_t \int_{\mathbf{x}} r_t(\mathbf{x}) g_{(1-t),t}(\mathbf{x}) p_t(\mathbf{x}) d\mathbf{x}$$

where $g_{01}(\mathbf{x}) = \int_{\mathbf{x}'} (r_0(\mathbf{x}') - r_0(\mathbf{x})) q_1(\mathbf{x}'|\mathbf{x}) d\mathbf{x}'$ and $g_{10}(\mathbf{x}) = \int_{\mathbf{x}'} (r_1(\mathbf{x}') - r_1(\mathbf{x})) q_0(\mathbf{x}'|\mathbf{x}) d\mathbf{x}'$. Here $g_{t',t}(\mathbf{x})$ denotes the expected gap between residuals of $\mathbf{x}$ and its neighbours. Please refer to Appendix section B.1 for the proof.

**Definition 5.7.** An Integral Probability Metric (IPM) between two probability distributions $p,q$ given a real-valued family of functions $G$ is defined as:

$$\text{IPM}_G(p,q) = \sup_{g \in G} \left| \int_{\mathbf{x}} g(\mathbf{x})(p(\mathbf{x}) - q(\mathbf{x})) d\mathbf{x} \right|$$

Examples of IPM include Maximum Mean Discrepancy and Wasserstein distance (Sriperumbudur et al., 2009).

**Assumption 5.8** (Expected distance to neighbors is bounded)**.** There exists a $\delta > 0$ such that $\int_{\mathbf{x}'} \|\mathbf{x} - \mathbf{x}'\|^2 q_t(\mathbf{x}'|\mathbf{x}) \leq \delta$.

**Theorem 5.9.** *We can now bound ITE Risk with PairNet Loss as:*

$$\epsilon_{ITE} \leq \epsilon_{pair} + \sum_t u_t \left[ B \cdot \text{IPM}_G(p_t,q_t) + 2K_{(1-t)} \delta \sqrt{\epsilon^t_F} \right]$$

*when we assume that the error residuals $r_0(\mathbf{x})$, $r_1(\mathbf{x})$ are $K_0$,$K_1$ Lipschitz respectively, and there exists a $B$ such that $\frac{1}{B}r_t(\mathbf{x})^2 \in G$, and the identifiability assumptions A1–A3 hold for $P(X,T,Y)$.*

Please refer to Appendix section B.2 for the proof.

Now, we derive conditions under which PairNet loss is a consistent estimator of ITE risk.

**Lemma 5.10** (Consistency of PairNet)**.** *Under the strict overlap assumption, PairNet is a consistent estimator of ITE.*

$$\lim_{N_t \to \infty} \epsilon_{ITE} = 0$$

Please refer to Appendix section B.3 for the proof.

Thus, for large enough $N_t$, minimising PairNet Loss drives the ITE Risk to zero.

### 5.1. Comparison with bounds of existing methods

We compare bounds on PairNet's ITE risk to those of the factual model. Theorem 1 of (Shalit et al., 2017) provides the bounds of the factual model as:

**Theorem 5.11.** *The ITE risk can be bounded with Factual loss as follows:*

$$\epsilon_{ITE} \leq 2(\epsilon_F^0 + \epsilon_F^1 + B \cdot IPM_G(p_1,p_0))$$

*when we assume that there exists a $B$ such that $\frac{1}{B}r_t(\mathbf{x})^2 \in G$, and the identifiability assumptions A1–A3 hold on $P(X,T,Y)$.*

Comparing the bounds of the factual loss above with PairNet loss in Thm 5.9 we make two remarks:

**Remark.** *First note that, the bound on ITE Risk with Factual loss does not go to zero even for large $N_t$ as the $\mathrm{IPM}_G(p_0,p_1)$ would be non-zero, in general, in the presence of confounding. In contrast, the bound with PairNet loss converges to zero under infinite samples as shown above.*

**Remark.** *Even under finite samples, the bounds of PairNet are significantly tighter than for factual loss. Even if we focus on the differences between the IPM terms across the two bounds, then $\mathrm{IPM}_G(p_0,p_1)$ is observed to be larger than $\sum_t u_t \mathrm{IPM}_G(p_t,q_t)$.*

As an illustration, we consider a toy setting where the densities $p_0,p_1$ are unit variance Gaussians with means $-1,+1$ respectively. We present the $p_t$ distribution and their induced neighbor distributions $q_t$ by PairNet in Figure 2. We observe that PairNet demonstrates a significantly lower MMD divergence of 0.09 compared to the factual model's 0.74. Notably, the divergence of PairNet would decrease further with increasing samples in the training dataset, while the factual model's divergence would remain the same. Even on real datasets we observe this trend as we discuss in the experiments section.

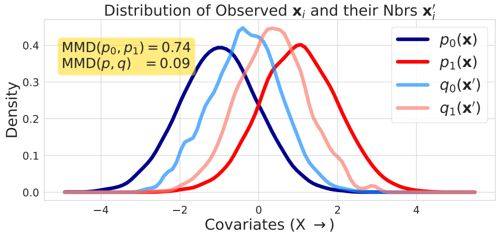


Figure 2: We plot the distributions $p_t$ and $q_t$, where $p_0$ is $\mathcal{N}(-1,1)$ and $p_1$ is $\mathcal{N}(+1,1)$. We observe that the factual model relying on MMD($p_0$,$p_1$) shows a larger difference from the ITE risk compared to PairNet, which depends on MMD($p$,$q$).

## 6. Empirical Evaluation

We conduct experiments to address the following research questions.

RQ1 Does PairNet outperform state-of-the-art methods?
RQ2 How sensitive is PairNet to the proximity of the pairs compared to matching methods that compute pseudo-outcomes from matched pairs?
RQ3 How aligned is the Pair loss with the ITE risk (Eq. 1) on real datasets?
RQ4 How sensitive is PairNet to the choice of hyper-parameters $\delta_{\text{pair}}$ and $\text{num}_{z'}$?
RQ5 How does Pair loss perform when applied on other T-Learners?

**Performance Metric.** We evaluate the Individual Treatment Effect (ITE) risk on a dataset $D_{\text{tst}}$, comprising counterfactual pairs represented as 5-tuples $(\mathbf{x},t,y,t',y')$. The Precision in Estimating Heterogeneous Effects (PEHE) (Johansson et al., 2016) serves as an empirical measure of the ITE risk, defined as: $\sqrt{\frac{1}{|D_{\text{tst}}|}\sum_{(\mathbf{x},t,y,t',y')}(\tau^*(\mathbf{x},t,t') - \hat{\tau}(\mathbf{x},t,t'))^2}$ We quantify PEHE (in) error for training instances and PEHE (out) error for test instances. Additionally, we employ hypothesis tests to evaluate the statistical significance of our results. Unlike conventional experiments that report standard deviation across runs, each seed in our experiments corresponds to a unique dataset, making hypothesis tests more appropriate, as noted in (Curth et al., 2021). We perform a one-sided paired t-test to compare PairNet performance with the baseline methods. A $p$-value below 0.05 indicates statistically significant performance improvements of PairNet over the baselines. $p$-values are enclosed in brackets in our tables.

**Experimental Setup** We implemented PairNet in the CATENets library [1] using JAX. For model training, all methods reserved 30% of the data for validation and implemented early stopping based on it. PairNet early stops on pairs as shown in Algorithm 1. We configured all hyperparameters, network architecture, optimizer, learning rate, etc. according

[1] https://github.com/AliciaCurth/CATENets

| | | IHDP | | ACIC | | Twins | |
|---|---|---|---|---|---|---|---|
| | | PEHE in | PEHE out | PEHE in | PEHE out | PEHE in | PEHE out |
| Meta-Learners | TLearner (Künzel et al., 2019) | 1.12 (0.00) | 1.34 (0.00) | 3.54 (0.02) | 4.29 (0.03) | 0.32 (0.25) | 0.32 (0.01) |
| | RLearner (Nie & Wager, 2021) | 3.14 (0.00) | 3.24 (0.00) | 4.01 (0.00) | 3.94 (0.00) | 0.32 (0.42) | 0.32 (0.15) |
| | DRLearner (Kennedy, 2020) | 1.12 (0.00) | 1.35 (0.00) | 3.01 (0.11) | 3.33 (0.08) | 0.32 (0.42) | 0.32 (0.14) |
| | XLearner (Künzel et al., 2019) | 1.77 (0.00) | 1.91 (0.00) | 2.90 (0.11) | 3.31 (0.10) | 0.32 (0.28) | 0.32 (0.01) |
| Rep. Learners | TARNet (Künzel et al., 2019) | 0.74 (0.01) | 0.83 (0.11) | 2.56 (0.26) | 2.71 (0.29) | 0.33 (0.01) | 0.32 (0.00) |
| | CFRNet (Johansson et al., 2016) | 0.96 (0.00) | 1.11 (0.00) | 3.12 (0.06) | 3.45 (0.06) | 0.33 (0.00) | 0.33 (0.00) |
| | FlexTENet (Curth & van der Schaar, 2021) | 1.03 (0.00) | 1.26 (0.00) | 3.78 (0.00) | 5.37 (0.00) | 0.37 (0.00) | 0.36 (0.00) |
| | ESCFR (Wang et al., 2024) | 0.99 (0.12) | 1.25 (0.22) | 2.84 (0.21) | 3.04 (0.29) | 0.33 (0.01) | 0.36 (0.00) |
| | StableCFR (Wu et al., 2023) | 1.48 (0.00) | 1.7 (0.02) | 4.39 (0.01) | 4.53 (0.02) | 0.36 (0.00) | 0.32 (0.00) |
| Weighting | IPW (Chesnaye et al., 2022) | 0.83 (0.00) | 0.93 (0.04) | 2.34 (0.44) | 2.57 (0.41) | 0.33 (0.00) | 0.33 (0.00) |
| | DragonNet (Shi et al., 2019) | 0.73 (0.01) | 0.83 (0.11) | 2.57 (0.25) | 2.72 (0.28) | 0.33 (0.01) | 0.33 (0.00) |
| Matching | $k$NN (Stuart, 2010) | 1.34 (0.00) | 1.46 (0.00) | 3.13 (0.03) | 3.14 (0.05) | 0.33 (0.13) | 0.32 (0.00) |
| | Perfect Match (Schwab et al., 2018) | 2.50 (0.00) | 2.66 (0.00) | 4.13 (0.00) | 4.02 (0.00) | 0.34 (0.00) | 0.33 (0.00) |
| | PairNet | 0.58 (0.00) | 0.69 (0.00) | 2.27 (0.00) | 2.46 (0.00) | 0.32 (0.00) | 0.32 (0.00) |

Table 1: RQ 1: The performance of PairNet compared to baselines evaluated using PEHE error on binary datasets. The table shows mean values and corresponding $p$-values within brackets. One-sided paired $t$-tests are conducted using PairNet as reference. Algorithms with the best mean error are highlighted in green, while those with large $p$-values are highlighted in yellow. Overall, PairNet demonstrates superior performance among all methods.

to the CATENets defaults. The sole change was setting the weight for the $L_2$ regularizer on the $\phi$ parameters to 1, a change applied uniformly to both PairNet and the baseline methods as it boosted the performance of all the models. Exclusive to PairNet are two hyperparameters: $\delta_{\text{pair}}$ and $\text{num}_{z'}$, set to 0.1 and 3, respectively, across all datasets. For $\psi$, which is used in distance computation during pair selection, we used embeddings from the representation network of a model trained until convergence on the factual loss.

### 6.1. RQ1: PairNet vs. Baselines

We address this question by comparing the performance of PairNet with baselines on both binary and continuous treatments. We begin our analysis with binary treatments.

**Binary Datasets.** We use the following three benchmark datasets: IHDP, ACIC, and Twins. The IHDP and ACIC datasets are semi-synthetic with synthetic potential outcome functions $\mu^*(\mathbf{x},t)$, while the Twins dataset contains real outcomes. We briefly describe them: **IHDP** The Infant Health Development Dataset (Johansson et al., 2016) contains 25 covariates, 747 examples, and 100 different realizations of the synthetic potential outcome function. **ACIC** The Atlantic Causal Inference Conference competition dataset (2016)[2] has 58 covariates, 4802 examples and considers 77 different potential outcome functions. Following CATENEts (Curth et al., 2021), we focused on three functions, namely versions 2, 7, 26, as they exhibit differences in levels of effect heterogeneity. In particular ACIC2 has no effect heterogeneity. **Twins** The Twins dataset (Louizos et al., 2017) provides ground truth outcomes for both treatments and has 40 covariates on 11,984 same-sex twins weighing less than 2kg at birth. The treatment variable indicates which twin in each pair is heavier. The outcome variable $Y$ is binary indicating the mortality within the first year of birth. Given the binary nature of outcomes ($Y$) in the Twins dataset, modeling the outcome difference requires a 3-way classification task, which we elaborate on in Appendix E. We defer a detailed description of the datasets to section C.1 in the Appendix.

**Binary Baselines.** We group the baselines based on how they address confounding. *Meta-learners* such as TLearner (Künzel et al., 2019), RLearner (Nie & Wager, 2021) directly learn Individual Treatment Effects (ITE) $\tau$ after imputing pseudo-outcomes for the missing treatments. *Representation-learning* methods like TARNet (Shalit et al., 2016), CFRNet (Shalit et al., 2017), FlexTENet (Curth & van der Schaar, 2021) share $\phi$ model parameters and learn treatment-specific $\mu_t$, with varied regularization. *Weighting* techniques like IPTW (Chesnaye et al., 2022) impose weighted factual losses. DragonNet (Shi et al., 2019) is a doubly robust method that is similar to the Augmented IPTW estimator. *Matching* approaches like $k$NN (Stuart, 2010) and Perfect Match (Schwab et al., 2018) perform pairing $(\mathbf{x},t,\mathbf{x}',t')$ like PairNet but copy over the outcome of $\mathbf{x}'$ as pseudo-outcomes for $\mathbf{x}$ under $t'$ and impose loss $(\mu(\mathbf{x},t')-y')^2$. The pairs have to be very close for such losses not to hurt.

**Results on Binary Treatments.** We present the results in Table 1 and emphasize the following key observations: **(1)** Overall, PairNet outperforms all eleven methods spanning all four categories of prior techniques for ITE. The gains on IHDP and ACIC are substantial and on Twins, all methods that model the outcome difference provide similar performance. **(2)** Meta Learners exhibit poor performance on

[2] https://jenniferhill7.wixsite.com/acic-2016/competition

| | IHDP | News | TCGA-0 | | TCGA-1 | | TCGA-2 | |
|---|---|---|---|---|---|---|---|---|
| Training Data size | $\lvert D\rvert$ | $\lvert D\rvert$ | $\lvert D\rvert$ | $0.1\times\lvert D\rvert$ | $\lvert D\rvert$ | $0.1\times\lvert D\rvert$ | $\lvert D\rvert$ | $0.1\times\lvert D\rvert$ |
| DRNet (Schwab et al., 2020) | 2.45 (0.00) | 1.42 (0.00) | 0.34 (0.01) | 0.52 (0.00) | 0.24 (0.04) | 0.27 (0.53) | 0.49 (0.44) | 0.77 (0.06) |
| PairNet (DRNet) | 2.27(0.00) | 1.32(0.00) | 0.25 (0.00) | 0.44 (0.00) | 0.21 (0.00) | 0.27 (0.00) | 0.48 (0.00) | 0.65 (0.00) |
| VCNet (Nie et al., 2021) | 1.73 (0.02) | 1.24(1.00) | 0.25 (0.57) | 0.43 (0.02) | 0.21 (0.38) | 0.27 (0.00) | 0.45 (0.51) | 0.58 (0.12) |
| PairNet (VCNet) | 1.57(0.00) | 1.26 (0.00) | 0.25 (0.00) | 0.27 (0.00) | 0.21 (0.00) | 0.22 (0.00) | 0.45 (0.00) | 0.49 (0.00) |

Table 2: RQ 1: Performance of Pair loss on DRNet, VCNet assessed using PEHE out error on continuous datasets. We report mean and $p$-values within brackets for a one-sided paired t-test conducted with PairNet as the baseline.

IHDP and ACIC due to their two-staged regression approach, where missing outcomes are imputed in the first stage. Errors from the first stage regression are propagated to the second stage, resulting in suboptimal $\hat{\tau}$. **(3)** Representation learners outperform meta-learners by joint training. However, these models lack the necessary inductive bias for predicting outcome differences across treatments during inference. ITE estimates can be particularly affected when error residuals for observed and alternative treatments exhibit a negative correlation. In contrast, PairNet promotes a positive correlation for nearby instances, leading to enhanced performance. **(4)** Weighting methods exhibit poor performance because they rely on propensity scores $\pi_t(\mathbf{x})$, which are often not well-calibrated. **(5)** Matching methods perform poorly as they impose counterfactual losses on pseudo-outcomes, which can be unreliable. In summary, PairNet strikes a balance by incorporating the necessary inductive biases for inferring $\tau$, while avoiding reliability issues by imposing only factual losses. Finally, we show the results for individual ACIC versions in Appendix I. **(6)** We explain the results on the two recent baselines, StableCFR and ESCFR as follows: StableCFR results are poor because they search for pairs by partitioning the high-dimensional covariate space and conduct matching there. On the contrary, methods such as PairNet that perform pairing in low dimensional embedding $\psi$ space tend to perform better. ESCFR relies on optimal transport for balancing the covariate representations across the treatment groups. However, OT based solutions suffer when implemented in a mini-batch on datasets with skewed treatment distributions.

**Continuous Experiments:** The treatments in continuous datasets take a real value between 0 and 1. **Continuous Datasets. TCGA[0-2]** (Bica et al., 2020) dataset obtained from The Cancer Genome Atlas project consists of 4,000 covariates that represent the gene expression of 9,659 cancer patients. We consider three different types of cancer treatments, and the outcome variable models the risk of cancer recurrence. **IHDP** (Johansson et al., 2016) covariates are the same as in the discrete case but treatments and outcomes are synthetic as proposed in (Nie et al., 2021). **News** (Bica et al., 2020) contains 2,858 bag-of-words covariates from 3,000 news articles taken from the New York Times. The treatment models the amount of time a user spends reading a news article while the outcomes model user satisfaction. We specify the functional forms of the assumed potential outcomes and the treatment assignment mechanism in Appendix C.3, and a detailed discussion in Appendix C.2.

**Sampling Pairs.** To create pairs for continuous treatments for an observed instance $(\mathbf{x}_i, t_i, y_i) \in D$, we adopt the approach mentioned in (Nagalapatti et al., 2024) to find near neighbors. We first sample $t' \in U[0, 1]$ from a uniform distribution. Then, we select pairs that are in close proximity to $\mathbf{x}_i$ from within a subset of $D$ defined as $\{(\mathbf{x}_j, t_j, y_j) \in D \mid |t_j - t'| < 0.05\}$; i.e., the $q_{t_i}$ distribution is defined on the filtered subset. The only change required in Algorithm 1 for continuous treatments is in line 4, where we calculate distances $d(\mathbf{x}, \mathbf{x}')$ for $\mathbf{x}'$ in the filtered subset.

**Continuous Baselines.** Unlike our method, most earlier methods discussed for binary treatment do not naturally generalize to the continuous case. For continuous treatments, two state-of-the-art model architectures are: DRNet (Schwab et al., 2020) and VCNet (Nie et al., 2021). Both these models share parameters in the $\phi$ layers and learn treatment-specific $\mu$ parameters. The key distinction lies in their approach to handling the treatment input. While DRNet applies binning on the $t$, VCNet employs a more sophisticated strategy of applying a spline basis expansion on $t$ and then learning of $\mu_t$ as a smooth function on it.

**Results on Continuous Treatments.** We compare the performance of the factual loss (Eq. 2) and Pair loss using DRNet and VCNet. We present the PEHE out errors here in table 2 and include the PEHE in errors in the Appendix section J. **(1)** We observe that Pair loss achieves the best mean performance across all the datasets for DRNet. **(2)** On the VCNet model, we find that Pair loss yield a statistically significant improvement on the IHDP dataset, while factual losses perform best on the news dataset. **(3)** For VCNet, both PairNet and factual approaches exhibit similar performance on the TCGA datasets, with some $p$-values close to $0.5$. VCNet's strong smoothness inductive bias enables factual losses to saturate performance given sufficient data (approximately 5.5k for TCGA). However, upon repeating experiments by randomly dropping about $90\%$ of the data, we observed statistically significant gains in PEHE error over the factual model. Notably, dropping data does not significantly affect PairNet's performance, whereas it deteriorates the Factual's performance.

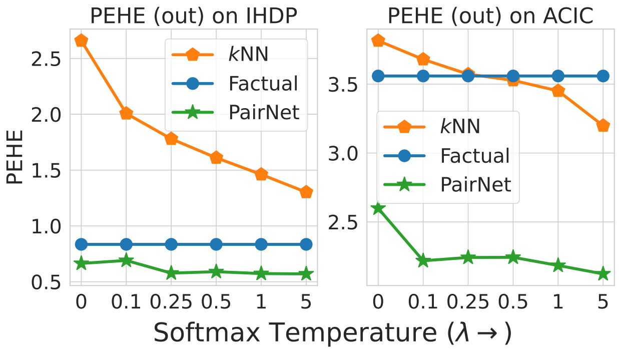


Figure 3: RQ2: PEHE with increasing proximity of covariates within a pair. Matching methods like $k$NN deteriorate fast if pairs are not close together, whereas PairNet remains robust and provides gains over baseline also with random pairing ($\lambda = 0$).

### 6.2. RQ2: Smaller Sensitivity of PairNet to pair proximity

We investigate this question by changing proximity via a temperature parameter $\lambda$ to softmax at step 6 of the `CreatePairs` function in Algorithm 1; that is, $q_{t_i}(\mathbf{x}_j|\mathbf{x}_i) \leftarrow \text{softmax}(-\lambda d_i)$. Setting $\lambda = 0$ gives us random pairs, whereas increasing $\lambda$ increases the proximity of pairs. We conduct experiments with IHDP and ACIC datasets for $\lambda$ across $\{0,0.1,0.25,0.5,1,5\}$. For reference, we compare (a) a Factual model that does not involve pairing and (b) $k$NN, which imposes counterfactual losses using paired instances. We present the results in Figure 1. We observe that the $k$NN method is adversely affected when provided with distant pairs (small $\lambda$). In contrast, PairNet demonstrates robustness. Interestingly, even with random pairs ($\lambda = 0$), PairNet performs better than the factual model because it aligns the error residuals for contrasting treatments.

We further conducted experiments with different choices for the embedding function $\psi$ using which the pairs are formed. We observed that PairNet performs well across a range of choices as shown in Section K in the Appendix.

### 6.3. RQ3: Alignment of Pair loss with ITE Risk

We assess the alignment on real datasets using two quantities: **(a)** correlation between Pair loss and the ITE risk, and **(b)** divergence between $p$ and $q$ distributions. These measures are computed on the test data and compared with the corresponding values achieved by the factual model. The results presented in Table 3 show that Pair loss exhibit a stronger correlation with the gold ITE risk. Further, we observe that PairNet achieves smaller MMD values even on real datasets.

### 6.4. RQ 4: Sensitivity of PairNet to $\delta_{\text{pair}}$ and num$_{z'}$

The parameter $\delta_{\text{pair}}$ denotes the proportion of pairs with the largest distance excluded in PairNet. Meanwhile, num$_{z'}$ indicates the number of pairs selected for each training sample. We vary $\delta_{\text{pair}}$ across the range $\{0,0.1,0.25\}$ and num$_{z'}$ across $\{1,2,3,4,5\}$. We assess the PEHE out errors and perform a t-test using PairNet default hyperparameter as the reference method. We emphasize the default value using cyan. The results for $\delta_{\text{pair}}$ are summarized in Table 4. Interestingly, we observe that PairNet demonstrates some sensitivity to the choice of this parameter in the continuous treatment setting, although this sensitivity is less pronounced in the binary case. Remarkably, PairNet consistently outperformed the baselines across all the considered parameter values.

|  | Correlation of ITE Risk Eq. 1 |  | Divergence |  |
|---|---|---|---|---|
|  | Factual Eq. 2 | PairNet Eq. 4 | MMD($p_0$,$p_1$) | MMD($p$,$q$) |
| IHDP | 0.748 | 0.807 | 0.120 | 0.100 |
| ACIC | 0.008 | 0.273 | 0.008 | 0.006 |

Table 3: RQ 3: Assessing alignment between Pair loss and the ITE risk. We observe that Pair loss exhibits a stronger correlation with the ITE risk when compared with the Factual model. Additionally, PairNet consistently achieves lower MMD measures.

|  | 0 | 0.25 | 0.1 |
|---|---|---|---|
| Binary | 0.48 (0.70) | 0.47 (0.62) | 0.45 (0.00) |
| Cont. (DRNet) | 2.26 (0.91) | 2.38 (0.61) | 2.40 (0.00) |
| Cont. (VCNet) | 1.41 (0.95) | 1.69 (0.36) | 1.63 (0.00) |

Table 4: RQ 4: PEHE out error for various values of $\delta_{\text{pair}}$ on binary and continuous versions of the IHDP dataset over 5 seeds.

We present sensitivity analysis results for different choices of num$_{z'}$ in Table 5. This hyperparameter governs the number of pairs created for each observed sample, with the default value set to 3 in PairNet. Table 5 demonstrates the robustness of PairNet to variations in num$_{z'}$ choices.

| IHDP | 1 | 2 | 4 | 5 | 3 |
|---|---|---|---|---|---|
| Binary | 0.50 (0.48) | 0.47 (0.67) | 0.47 (0.70) | 0.44 (0.84) | 0.50 (0.00) |
| Cont. (DRNet) | 2.31 (0.35) | 2.29 (0.42) | 2.21 (0.69) | 2.27 (0.47) | 2.26 (0.00) |
| Cont. (VCNet) | 1.46 (0.24) | 1.45 (0.28) | 1.45 (0.32) | 1.42 (0.47) | 1.41 (0.00) |

Table 5: PEHE out error for various values of num$_{z'}$ on both the binary and continuous versions of the IHDP dataset. PairNet performance is similar across all the five choices.

### 6.5. RQ5: Pair Loss on other T-Learners

Since Pair Loss is model-agnostic, it applies to any T-Learner-based model architecture. We chose TARNet as the default architecture due to its simplicity and lack of assumptions. In this experiment, we applied Pair Loss to other T-Learners, including CFRNet, DragonNet, and FlexTENet. The results, shown in Table 6 for IHDP and all three versions of the ACIC dataset, demonstrate that Pair Loss consistently improves performance, making it a preferable option across various model architectures.

| | CFRNet | | DragonNet | | FlexTENet | |
|---|---|---|---|---|---|---|
| Dataset | Baseline | PairNet | Baseline | PairNet | Baseline | PairNet |
| IHDP | 1.68 (0.33) | 1.03 (0.00) | 1.23 (0.46) | 0.90 (0.00) | 1.57 (0.44) | 1.20 (0.00) |
| ACIC2 | 2.12 (0.20) | 0.72 (0.00) | 1.15 (0.34) | 1.2 (0.00) | 4.89 (0.08) | 2.86 (0.00) |
| ACIC7 | 4.12 (0.01) | 3.55 (0.00) | 3.49 (0.06) | 2.90 (0.00) | 4.62 (0.12) | 3.85 (0.00) |
| ACIC26 | 4.12 (0.1) | 3.28 (0.00) | 3.52 (0.3) | 3.32 (0.00) | 6.60 (0.11) | 4.62 (0.00) |

Table 6: RQ 5: Performance of T-Learners, including CFRNet, DragonNet, and FlexTENet, when trained with Pair loss. We compare the results of the baseline losses with our approach and observe that Pair loss provides a significant performance improvement.

| $\alpha$ | 0 | 0.5 | 1 | 1.5 | 2 |
|---|---|---|---|---|---|
| IHDP | 0.89 (0.06) | 1.03 (0.01) | 1.03 (0.01) | 1.03 (0.01) | 0.69 (0.00) |
| ACIC2 | 0.92 (0.83) | 0.47 (0.97) | 1.5 (0.53) | 0.51 (0.96) | 1.56 (0.00) |
| ACIC7 | 2.68 (0.68) | 3.28 (0.19) | 2.89 (0.50) | 3.25 (0.20) | 2.89 (0.00) |
| ACIC26 | 2.72 (0.63) | 2.93 (0.51) | 2.77 (0.61) | 2.9 (0.53) | 2.95 (0.00) |

Table 7: This table shows the results of weighting the residual term in Pair loss with $\alpha$. PairNet's default choice is not to weight this term (i.e., $\alpha = 2$). We observe that no particular $\alpha$ achieves the best performance across all datasets with statistical significance.

| $\tau$ is | (a) Simpler | (b) Comparable | (c) Complex |
|---|---|---|---|
| Factual | 6.34 (0.00) | 7.73 (0.01) | 10.08 (0.07) |
| PairNet | 5.54 (0.00) | 6.22 (0.00) | 08.19 ( 0.00) |

Table 8: PEHE out errors for various complexities of the ITE function $\tau$ on synthetic data; PairNet outperforms Factual.

### 6.6. Additional Sensitivity Analysis

**Weighting the Residual Term in Pair Loss** We conducted an experiment where we adjusted the weight of the residual alignment term in Pair loss. Suppose we expand Pair loss as $(y-\hat{y})^2+(y'-\hat{y'})^2-\alpha(y-\hat{y})(y'-\hat{y'})$, with $\alpha=2$ as used in our paper. For Pair Loss to be non-negative we need $\alpha \in [0,2]$. We varied $\alpha$ across $\{0,0.5,1,1.5,2\}$. The p-values represent t-tests conducted with $\alpha=2$ as the baseline.

We show the results in Table 7, where we observed that no specific $\alpha$ value consistently outperformed others across the datasets. Additionally, many p-values were large, indicating that we couldn't draw conclusions on which $\alpha$ was superior with statistical significance. While we could have tuned $\alpha$ using pair error on a validation dataset, we opted to avoid hyper-parameter search and stuck to the simple squared error proposal as outlined in our paper.

**Performance across $\tau$ Complexities** We conducted an experiment with the synthetic dataset proposed in (Curth & van der Schaar, 2021). The dataset comprises three variants where $\tau$ is (a) simpler, (b) comparable, and (c) more complex than each of the $\mu_0, \mu_1$. Our results in table 8 show that PairNet consistently outperforms the factual model across $\tau$ complexities.

**Impact of $L_2$ Penalty** In Table 9, we illustrate the significance of applying an $L_2$ penalty to the $\phi$ parameters. We explore two different magnitudes of imposing the $L_2$ loss: 1e-4, the default value for CATENets, and a value of 1. We observed consistent enhancements across all the baselines, including PairNetby introducing a large penalty to the $\phi$ parameters. This improvement was particularly noticeable for PairNet with version 2 of the ACIC dataset. For a detailed analysis of this hyperparameter, please refer to Appendix Section G.

| | TARNet | | PairNet | |
|---|---|---|---|---|
| | 1e-4 | 1 | 1e-4 | 1 |
| IHDP | 1.51 (1.00) | 0.83 (0.00) | 0.74 (0.35) | 0.69 (0.00) |
| ACIC(2) | 3.05 (0.74) | 2.71 (0.00) | 3.52 (1.00) | 2.46 (0.00) |

Table 9: PEHE out error for various values of $L_2$ penalty applied on the $\phi$ parameters. A weight of 1 achieves better results.

## 7. Conclusion

In this paper, we introduced PairNet, a simple yet effective approach for estimating treatment effects. PairNet is model-agnostic, applicable to diverse treatment domains, and can be integrated with most prior networks. Our key idea was to only impose factual losses on pairs of neighboring instances. We showed that this approach effectively aligns error residuals for the chosen pairs, thereby aiding in ITE inference when outcomes are predicted for the same instance under two treatments. Across various benchmarks involving both discrete and continuous treatments, PairNet showcased significant improvements over most existing methods. Furthermore, we theoretically characterized the difference between Pair loss and the ITE risk, showing that this difference depends on the proximity between instances in the selected pairs. This implies that under certain overlap assumptions, given a large training dataset, Pair loss serves as a consistent estimate of the ITE risk. Additionally, through several experiments and sensitivity analyses, we highlighted the merits of PairNet and provided insight into the superior performance achieved by PairNet over the baselines.

## Acknowledgement

Lokesh acknowledges funding from the Prime Minister Research Fellowship, Government of India, and Microsoft Research India for the travel award.

## Impact Statement

Developing new and effective methods like PairNet to understand how treatments impact different situations is crucial for decision-making across various fields like healthcare, economics, and education. For instance, in healthcare, precise knowledge about how treatments affect individuals can lead to better, more personalized medical care, ultimately improving access to quality healthcare. Similarly, in economics, understanding the consequences of policies can help policymakers make informed decisions that benefit society.

However, while these methods provide valuable insights, it is crucial to address potential ethical concerns, particularly regarding fairness. Before deploying these methods, it is essential to ensure that the observational dataset used is unbiased and does not unfairly benefit certain groups of individuals, which could lead to biased effect prediction during inference. Addressing such issues is essential for the responsible deployment of these methods and their positive impact on decision-making and societal outcomes.

## References

Alaa, A. M. and Van Der Schaar, M. Bayesian inference of individualized treatment effects using multi-task gaussian processes. *Advances in neural information processing systems*, 30, 2017.

Alaa, A. M. and van der Schaar, M. Deep multi-task gaussian processes for survival analysis with competing risks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 2326–2334, 2017.

Bellot, A., Dhir, A., and Prando, G. Generalization bounds and algorithms for estimating conditional average treatment effect of dosage. *arXiv preprint arXiv:2205.14692*, 2022.

Bica, I., Jordon, J., and van der Schaar, M. Estimating the effects of continuous-valued interventions using generative adversarial networks. *Advances in Neural Information Processing Systems*, 33:16434–16445, 2020.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Chauhan, V. K., Molaei, S., Tania, M. H., Thakur, A., Zhu, T., and Clifton, D. A. Adversarial de-confounding in individualised treatment effects estimation. In *International Conference on Artificial Intelligence and Statistics*, pp. 837–849. PMLR, 2023.

Chesnaye, N. C., Stel, V. S., Tripepi, G., Dekker, F. W., Fu, E. L., Zoccali, C., and Jager, K. J. An introduction to inverse probability of treatment weighting in observational research. *Clinical Kidney Journal*, 15(1):14–20, 2022.

Curth, A. and van der Schaar, M. On inductive biases for heterogeneous treatment effect estimation. *Advances in Neural Information Processing Systems*, 34:15883–15894, 2021.

Curth, A. and van der Schaar, M. In search of insights, not magic bullets: Towards demystification of the model selection dilemma in heterogeneous treatment effect estimation. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, 2023.

Curth, A., Svensson, D., Weatherall, J., and van der Schaar, M. Really doing great at estimating cate? a critical look at ml benchmarking practices in treatment effect estimation. In *Thirty-fifth conference on neural information processing systems datasets and benchmarks track (round 2)*, 2021.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

D'Amour, A., Ding, P., Feller, A., Lei, L., and Sekhon, J. Overlap in observational studies with high-dimensional covariates. *Journal of Econometrics*, 221(2):644–654, 2021.

Gao, Z. and Han, Y. Minimax optimal nonparametric estimation of heterogeneous treatment e ffects. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and in, H. L. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21751–21762. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/f75b757d3459\c3e93e98ddab7b903938-Paper.pdf.

Hassanpour, N. and Greiner, R. Counterfactual regression with importance sampling weights. In *IJCAI*, pp. 5880–5887, 2019a.

Hassanpour, N. and Greiner, R. Learning disentangled representations for counterfactual regression. In *International Conference on Learning Representations*, 2019b.

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

Iacus, S. M., King, G., and Porro, G. Causal inference without balance checking: Coarsened exact matching. *Political analysis*, 20(1):1–24, 2012.

Johansson, F., Shalit, U., and Sontag, D. Learning representations for counterfactual inference. In *International conference on machine learning*, pp. 3020–3029. PMLR, 2016.

Johnson, J., Douze, M., and Jégou, H. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

Jung, Y., Tian, J., and Bareinboim, E. Learning causal effects via weighted empirical risk minimization. *Advances in neural information processing systems*, 33:12697–12709, 2020.

Kallus, N. Deepmatch: Balancing deep covariate representations for causal inference using adversarial training. In *International Conference on Machine Learning*, pp. 5067–5077. PMLR, 2020.

Kennedy, E. H. Towards optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*, 2020.

Künzel, S. R., Sekhon, J. S., Bickel, P. J., and Yu, B. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.

Lu, D., Tao, C., Chen, J., Li, F., Guo, F., and Carin, L. Reconsidering generative objectives for counterfactual reasoning. *Advances in Neural Information Processing Systems*, 33:21539–21553, 2020.

Nagalapatti, L., Iyer, A., De, A., and Sarawagi, S. Continuous treatment effect estimation using gradient interpolation and kernel smoothing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):14397–14404, Mar. 2024. doi: 10.1609/aaai.v38i13.29353. URL https://ojs.aaai.org/index.php/AAAI/article/view/29353.

Nie, L., Ye, M., Liu, Q., and Nicolae, D. Vcnet and functional targeted regularization for learning causal effects of continuous treatments. *arXiv preprint arXiv:2103.07861*, 2021.

Nie, X. and Wager, S. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, 108(2):299–319, 2021.

Ozery-Flato, M., Thodoroff, P., and El-Hay, T. Adversarial balancing for causal inference. *ArXiv*, abs/1810.07406, 2018.

Rissanen, S. and Marttinen, P. A critical look at the consistency of causal estimation with deep latent variable models. *Advances in Neural Information Processing Systems*, 34:4207–4217, 2021.

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

Rubin, D. B. Causal Inference Using Potential Outcomes: Design, Modeling, Decisions. *Journal of the American Statistical Association*, 100:322–331, March 2005. URL https://ideas.repec.org/a/bes/jnlasa/v100y2005p322-331.html.

Schwab, P., Linhardt, L., and Karlen, W. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. *arXiv preprint arXiv:1810.00656*, 2018.

Schwab, P., Linhardt, L., Bauer, S., Buhmann, J. M., and Karlen, W. Learning counterfactual representations for estimating individual dose-response curves. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5612–5619, 2020.

Shalit, U., Johansson, F. D., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms, 2016. URL https://arxiv.org/abs/1606.03976.

Shalit, U., Johansson, F. D., and Sontag, D. Estimating individual treatment effect: generalization bounds and algorithms. In *International Conference on Machine Learning*, pp. 3076–3085. PMLR, 2017.

Shi, C., Blei, D., and Veitch, V. Adapting neural networks for the estimation of treatment effects. *Advances in neural information processing systems*, 32, 2019.

Sriperumbudur, B. K., Fukumizu, K., Gretton, A., Schölkopf, B., and Lanckriet, G. R. G. On integral probability metrics, $\phi-divergences and binary classification,$2009.

Stuart, E. A. Matching methods for causal inference: A review and a look forward. *Statistical science: a review journal of the Institute of Mathematical Statistics*, 25(1):1, 2010.

Wager, S. and Athey, S. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

Wang, H., Fan, J., Chen, Z., Li, H., Liu, W., Liu, T., Dai, Q., Wang, Y., Dong, Z., and Tang, R. Optimal transport for treatment effect estimation. *Advances in Neural Information Processing Systems*, 36, 2024.

Wu, A., Kuang, K., Xiong, R., Li, B., and Wu, F. Stable estimation of heterogeneous treatment effects. In *International Conference on Machine Learning*, pp. 37496–37510. PMLR, 2023.

Yao, L., Li, S., Li, Y., Huai, M., Gao, J., and Zhang, A. Representation learning for treatment effect estimation from observational data. *Advances in Neural Information Processing Systems*, 31, 2018.

Yoon, J., Jordon, J., and Van Der Schaar, M. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

Zhang, Y., Bellot, A., and Schaar, M. Learning overlapping representations for the estimation of individualized treatment effects. In *International Conference on Artificial Intelligence and Statistics*, pp. 1005–1014. PMLR, 2020.

Zhang, Y.-F., Zhang, H., Lipton, Z. C., Li, L. E., and Xing, E. P. Exploring transformer backbones for heterogeneous treatment effect estimation, 2022. URL https://arxiv.org/abs/2202.01336.

# Appendix

**(PairNet: Training with Observed Pairs to Estimate Individual Treatment Effect)**

## A. Explanation for PairNet Algorithm

We explain the pseudocode in Alg. 1 as follows:

- At line 2, the training dataset is split into $D_{\text{trn}}$ and $D_{\text{val}}$, stratified by treatments. For continuous $T$, a random split is used.
- At line 3, pairs are created to compute validation performance for early stopping. It is important to note that pairs for samples in the validation dataset can also be obtained from $D_{\text{trn}}$. This approach is taken because $D_{\text{val}}$ is sparse, and searching for pairs within it may result in many distant pairs, which would not accurately reflect true validation performance.
- During each epoch $e$, pairs ($D^e_{trn}$) for imposing pair loss are created (line 5) using the CreatePairDS procedure. For training, pairs are created within the $D_{\text{trn}}$ split.
- The CreatePairDS procedure samples $num_{z'}$ neighbors based on the $q_t$ distribution, defined as a softmax over negative distances (line 6).
- Line 10 includes $\delta_{\text{pair}}$ to avoid imposing losses on extremely distant pairs. Training continues using standard mini-batch gradient descent on Pair loss (lines 7-10).
- Once paired samples for training are obtained, the pair loss is imposed, and gradient descent is performed until convergence, monitored by the pair loss on $D_{\text{val}}$.

**Intuition:.** The main intuition behind PairNet is to better correlate with ITE loss by fitting the difference in outcomes of nearby covariates. This is further explained using Equation 4 and Figure 1. We argue that Pair Loss aligns better with ITE loss because its ITE generalization gap (Theorem 5.9) is bounded by $D(p_t(X)||q_t(X))$, where $p_t(X)$ represents observed covariates for $T=t$ and $q_t(X)$ represents paired neighbor distribution. This is much lower than that of a factual model, bounded by $D(p_0||p_1)$, as depicted in Figure 2 and Thm 5.12.

## B. Theoretical Analysis

**Remark.** *For ease of notation, in the main paper, we showed results when the representation extraction $\phi$ is an identity map. In the Appendix, we present a generalized version of our results under certain assumptions on $\phi$ which we elaborate on now. Note that this modification does not affect the form of bounds that we obtain for Factual or PairNet.*

**Assumption B.1** ($\phi$ is diffeomorphic)**.** The embedding function $\phi$ is a twice-differentiable, invertible function, serving as a push-forward operator between the spaces $\mathcal{X}$ and $\Phi$.

**Definition B.2** (Covariate distribution under $\phi$)**.** Under the push-forward operator $\phi$, for an instance $e=\phi(\mathbf{x})$, we define $p^{\phi}_t(e)=p_t(\phi^{-1}(e))\left|\frac{d\phi^{-1}(e)}{de}\right|$.

**Assumption B.3** (Instance error lies in $G$)**.** Let $G$ denote a family of functions $\{g:\Phi\mapsto\mathbb{R}\}$. There exists a constant $B_\phi$ such that for any $e=\phi(\mathbf{x}),t$, we have $\frac{1}{B_\phi}r_t(\phi^{-1}(e))^2\in G$, where $r_t(\mathbf{x})^2$ is the squared error.

In the following proofs, IPM$_G$ is modified to consider a function space $G$ over $\Phi$ instead of $\mathcal{X}$ in the main paper. Thus all bounds with $B\cdot$IPM$_G(p_t,q_t)$ change to $B_\phi\cdot$IPM$_G(p^\phi_t,q^\phi_t)$

**Remark.** *The original bounds in (Shalit et al., 2017) features a $\sigma^2_Y$ term that stems from the irreducible noise in the observed outcomes $y_i$. However, we drop that term in our analysis for brevity.*

### B.1. Proof of Lemma 5.6

The difference between ITE Risk and PairNet loss can be expressed as

$$\begin{aligned}\epsilon_{\text{ITE}}-\epsilon_{\text{pair}}=&\sum_t u_{(1-t)}\int_{\mathbf{x}} r_t(\mathbf{x})^2(p_t(\mathbf{x})-q_t(\mathbf{x}))d\mathbf{x}\\&+\sum_t 2u_t\int_{\mathbf{x}} r_t(\mathbf{x})g_{(1-t),t}(\mathbf{x})p_t(\mathbf{x})d\mathbf{x}\end{aligned}$$

**Proof.**

$$\begin{aligned}
\epsilon_{\text{ITE}} &= \int_{\mathbf{x}} (r_1(\mathbf{x}) - r_0(\mathbf{x}))^2 p(\mathbf{x}) d\mathbf{x} \\
&= \int_{\mathbf{x}} (r_1(\mathbf{x}) - r_0(\mathbf{x}))^2 (p_0(\mathbf{x}) u_0 + p_1(\mathbf{x}) u_1) d\mathbf{x} = u_0 \int_{\mathbf{x}} (r_1(\mathbf{x}) - r_0(\mathbf{x}))^2 p_0(\mathbf{x}) d\mathbf{x} + u_1 \int_{\mathbf{x}} (r_1(\mathbf{x}) - r_0(\mathbf{x}))^2 p_1(\mathbf{x}) d\mathbf{x} \\
&= u_0 \int_{\mathbf{x}} (r_1(\mathbf{x})^2 + r_0(\mathbf{x})^2 - 2 r_1(\mathbf{x}) r_0(\mathbf{x})) p_0(\mathbf{x}) d\mathbf{x} + u_1 \int_{\mathbf{x}} (r_1(\mathbf{x})^2 + r_0(\mathbf{x})^2 - 2 r_1(\mathbf{x}) r_0(\mathbf{x})) p_1(\mathbf{x}) d\mathbf{x} \\
\epsilon_{\text{pair}} &= \int_{\mathbf{x}} \int_{\mathbf{x}'} (r_1(\mathbf{x}) - r_0(\mathbf{x}'))^2 q_1(\mathbf{x}'|\mathbf{x}) p_1(\mathbf{x}) u_1 d\mathbf{x} d\mathbf{x}' + \int_{\mathbf{x}} \int_{\mathbf{x}'} (r_1(\mathbf{x}') - r_0(\mathbf{x}))^2 q_0(\mathbf{x}'|\mathbf{x}) p_0(\mathbf{x}) u_0 d\mathbf{x} d\mathbf{x}' \\
&= \int_{\mathbf{x}} \int_{\mathbf{x}'} (r_1(\mathbf{x})^2 + r_0(\mathbf{x}')^2 - 2 r_1(\mathbf{x}) r_0(\mathbf{x}')) q_1(\mathbf{x}'|\mathbf{x}) p_1(\mathbf{x}) u_1 d\mathbf{x} d\mathbf{x}' \\
&+ \int_{\mathbf{x}} \int_{\mathbf{x}'} (r_1(\mathbf{x}')^2 + r_0(\mathbf{x})^2 - 2 r_1(\mathbf{x}') r_0(\mathbf{x})) q_0(\mathbf{x}'|\mathbf{x}) p_0(\mathbf{x}) u_0 d\mathbf{x} d\mathbf{x}' \\
&= \int_{\mathbf{x}} r_1(\mathbf{x})^2 p_1(\mathbf{x}) u_1 d\mathbf{x} + \int_{\mathbf{x}'} r_0(\mathbf{x}')^2 q_1(\mathbf{x}') u_1 d\mathbf{x}' - \int_{\mathbf{x}} \int_{\mathbf{x}'} 2 r_1(\mathbf{x}) r_0(\mathbf{x}') q_1(\mathbf{x}'|\mathbf{x}) p_1(\mathbf{x}) u_1 d\mathbf{x} d\mathbf{x}' \\
&\quad \text{(By marginalizing over variables)} \\
&+ \int_{\mathbf{x}} r_0(\mathbf{x})^2 p_0(\mathbf{x}) u_0 d\mathbf{x} + \int_{\mathbf{x}'} r_1(\mathbf{x}')^2 q_0(\mathbf{x}') u_0 d\mathbf{x}' - \int_{\mathbf{x}} \int_{\mathbf{x}'} 2 r_1(\mathbf{x}') r_0(\mathbf{x}) q_0(\mathbf{x}'|\mathbf{x}) p_0(\mathbf{x}) u_0 d\mathbf{x} d\mathbf{x}'
\end{aligned}$$

Now taking the difference between both expressions and cancelling common terms, we obtain the lemma.

### B.2. Proof of Theorem 5.9

We can now bound ITE Risk with PairNet Loss as:

$$\epsilon_{\text{ITE}} \le \epsilon_{\text{pair}} + \sum_t u_t \left[ B \cdot \text{IPM}_G(p_t, q_t) + 2 K_{(1-t)} \delta \sqrt{\epsilon_{\text{F}}^t} \right]$$

when we assume that the error residuals $r_0(\mathbf{x})$, $r_1(\mathbf{x})$ are $K_0, K_1$ Lipschitz respectively, and there exists a $B$ such that $\frac{1}{B} r_t(\mathbf{x})^2 \in G$, and the identifiability assumptions A1–A3 hold for $P(X,T,Y)$.

**Proof.** From Lemma 5.6, we have

$$\begin{aligned}
\epsilon_{\text{ITE}} - \epsilon_{\text{pair}} = u_0 \int_{\mathbf{x}} r_1(\mathbf{x})^2 (p_0(\mathbf{x}) - q_0(\mathbf{x})) d\mathbf{x} + u_1 \int_{\mathbf{x}} r_0(\mathbf{x})^2 (p_1(\mathbf{x}) - q_1(\mathbf{x})) d\mathbf{x} \\
+ 2 u_1 \int_{\mathbf{x}} r_1(\mathbf{x}) g_{01}(\mathbf{x}) p_1(\mathbf{x}) d\mathbf{x} + 2 u_0 \int_{\mathbf{x}} r_0(\mathbf{x}) g_{10}(\mathbf{x}) p_0(\mathbf{x}) d\mathbf{x}
\end{aligned}$$

$$\begin{aligned}
\epsilon_{\text{ITE}} \le \epsilon_{\text{pair}} + \underbrace{u_0 \left| \int_{\mathbf{x}} r_1(\mathbf{x})^2 (p_0(\mathbf{x}) - q_0(\mathbf{x})) d\mathbf{x} \right| + u_1 \left| \int_{\mathbf{x}} r_0(\mathbf{x})^2 (p_1(\mathbf{x}) - q_1(\mathbf{x})) d\mathbf{x} \right|}_{\text{Term 1}} \\
\underbrace{+ 2 u_1 \left| \int_{\mathbf{x}} r_1(\mathbf{x}) g_{01}(\mathbf{x}) p_1(\mathbf{x}) d\mathbf{x} \right| + 2 u_0 \left| \int_{\mathbf{x}} r_0(\mathbf{x}) g_{10}(\mathbf{x}) p_0(\mathbf{x}) d\mathbf{x} \right|}_{\text{Term 2}}
\end{aligned}$$

Using $\frac{1}{B_\phi} r_t(\mathbf{x})^2 \in G$, we can bound Term 1 as follows:

$$\begin{aligned}
\left| \int_{\mathbf{x}} r_1(\mathbf{x})^2 (p_0(\mathbf{x}) - q_0(\mathbf{x})) d\mathbf{x} \right| &= \left| B_\phi \int_\Phi \frac{1}{B_\phi} r_1(\phi^{-1}(e))^2 (p_0^\phi(e) - q_0^\phi(e) de \right| \\
&\le B_\phi \sup_{g \in G} \left| \int_\Phi (g(e)(p_0^\phi(e) - q_0^\phi(e)) de \right| = B_\phi \text{IPM}_G(p_0^\phi, q_0^\phi)
\end{aligned}$$

Thus, Term 1 can be upper-bounded by $u_0 B_\phi \text{IPM}_G(p_0^\phi, q_0^\phi) + u_1 B_\phi \text{IPM}_G(p_1^\phi, q_1^\phi)$.

Recall that $g_{01}(\mathbf{x}) = \int_{\mathbf{x}'} (r_0(\mathbf{x}') - r_0(\mathbf{x})) q_1(\mathbf{x}'|\mathbf{x}) d\mathbf{x}'$ and $g_{10}(\mathbf{x}) = \int_{\mathbf{x}'} (r_1(\mathbf{x}') - r_1(\mathbf{x})) q_0(\mathbf{x}'|\mathbf{x}) d\mathbf{x}'$.

We now derive an upper bound for Term 2 as follows:

$$\begin{aligned}\left|\int_{\mathbf{x}} r_1(\mathbf{x}) g_{01}(\mathbf{x}) p_1(\mathbf{x}) d\mathbf{x}\right|^2 &\leq \int_{\mathbf{x}} r_1^2(\mathbf{x}) p_1(\mathbf{x}) d\mathbf{x} \int_{\mathbf{x}} (g_{01}(\mathbf{x}))^2 p_1(\mathbf{x}) d\mathbf{x} \text{ (By Cauchy-Schwarz Inequality)} \\ &\leq \int_{\mathbf{x}} r_1^2(\mathbf{x}) p_1(\mathbf{x}) d\mathbf{x} \int_{\mathbf{x}} (K_0^2 \delta^2) p_1(\mathbf{x}) d\mathbf{x} \\ &= \epsilon_{\mathrm{F}}^1 K_0^2 \delta^2\end{aligned}$$

Here we use the fact that $g_{t',t}(\mathbf{x})$ can be bounded as follows under the Lipschitz continuity assumption

$$|g_{01}(\mathbf{x})| \leq K_0 \delta \quad |g_{10}(\mathbf{x})| \leq K_1 \delta$$

$$|g_{01}(\mathbf{x})| = \left|\int_{\mathbf{x}'} (r_0(\mathbf{x}') - r_0(\mathbf{x})) q_1(\mathbf{x}'|\mathbf{x}) d\mathbf{x}'\right| \leq \int_{\mathbf{x}'} |(r_0(\mathbf{x}') - r_0(\mathbf{x}))| q_1(\mathbf{x}'|\mathbf{x}) d\mathbf{x}'$$

Using Lipschitz continuity $|(r_0(\mathbf{x}') - r_0(\mathbf{x}))| \leq K_0 \|\mathbf{x}' - \mathbf{x}\|$, $|(r_1(\mathbf{x}') - r_1(\mathbf{x}))| \leq K_1 \|\mathbf{x}' - \mathbf{x}\|$

$$|g_{01}(\mathbf{x})| \leq \int_{\mathbf{x}'} K_0 \|\mathbf{x}' - \mathbf{x}\| q_1(\mathbf{x}'|\mathbf{x}) d\mathbf{x}'$$

Now using 5.8, the expected neighbour distance is bounded by $\delta$ yielding $|g_{01}(\mathbf{x})| \leq K_0 \delta$. By symmetric arguments we can bound $|g_{10}(\mathbf{x})| \leq K_1 \delta$

Thus, $\left|\int_{\mathbf{x}} r_1(\mathbf{x}) g_{01}(\mathbf{x}) p_1(\mathbf{x}) d\mathbf{x}\right| \leq K_0 \delta \sqrt{\epsilon_{\mathrm{F}}^1}$. We can now bound Term 2 with $2u_1 K_0 \delta \sqrt{\epsilon_{\mathrm{F}}^1} + 2u_0 K_1 \delta \sqrt{\epsilon_{\mathrm{F}}^0}$

### B.3. Proof of Lemma 5.10

Under the strict overlap assumption, PairNet is a consistent estimator of ITE.

$$\lim_{N_t \to \infty} \epsilon_{\text{ITE}} = 0$$

**Proof.** We prove the consistency of PairNet under strict overlap (D'Amour et al., 2021). First, we show that strict overlap can be used to derive a lower bound on the radius of a sphere around any point in which one can find a sample with the opposite treatment with high probability. Next, we show that this lower bound decreases with number of samples, $m$, allowing us to shrink the radius in which we find neighbours, thereby ensuring $q_t$ distribution converges to $p_t$.

Strict overlap states that $c < p(t|x) < 1 - c$ for some $c > 0$. Consider any sample $x$ with treatment $t$. For pairing, we need to find a matching sample $x'$ with alternative treatment $t' = 1 - t$ from the $m$ observations in the dataset that are assigned treatment $t'$.

Define $\bar{p}_r = \int_{\mathbf{x}' \in \mathfrak{B}_r(\mathbf{x})} p(\mathbf{x}'|t') d\mathbf{x}'$ as the probability mass of $p(X = \mathbf{x}'|T = t')$ in a ball of radius $r$ around $x$ where $\mathfrak{B}_r(\mathbf{x}) := \{\mathbf{x}'|\mathbf{x}' \in \mathbb{R}^d, ||\mathbf{x}' - \mathbf{x}|| < r\}$.

The probability that at least one sample $\mathbf{x}_j'$ lies within this ball is $\bar{p}(r,m) = 1 - (1 - \bar{p}_r)^m$. If $\lim_{m \to \infty} \bar{p}(r,m) = 1$, then PairNet would find a near neighbor within this ball. This is satisfied for $\bar{p}_r = \omega(m^{-1})$ (where $\omega$ denotes asymptotic lower bound).

Now, we construct a condition for $\bar{p}_r = \omega(m^{-1})$. First note that $\bar{p}_r >$ Volume$(\mathfrak{B}_r(x)) \cdot \min_x' p(x'|t')$ (where the minimum is over the $x' \in \mathfrak{B}_r(x)$) and Volume$(\mathfrak{B}_r(x)) \propto r^d = kr^d$.

$\min_{x' \in \mathfrak{B}_r(x)} p(x'|t') = \min_{x' \in \mathfrak{B}_r(x)} \frac{p(t'|x') p(x')}{p(t')} > \frac{c}{p(t')} \min_{x' \in \mathfrak{B}_r(x)} p(x')$ (By Strict Overlap)

Combining these two, we get $\bar{p}_r > \frac{kc}{p(t')} r^d \min_{x' \in \mathfrak{B}_r(x)} p(x')$

If we shrink the ball around $x$ as $r = \omega(m^{-1/d}), \lim_{m \to \infty} r = 0$ then we get the desired condition $\bar{p}_r = \omega(m^{-1})$.

Therefore, for any $\mathbf{x}, t$ we can sample a neighbour $\mathbf{x}', t' = 1 - t$ within $\mathfrak{B}_r(x)$ for $r = \omega(m^{-1/d})$ with high probability $\bar{p}(r,m)$.

Since we set $r = \omega(m^{-1/d})$ which decreases with $m$, we have $\lim_{m \to \infty} r = 0$ and $\lim_{m \to \infty} \bar{p}(r,m) = 1$. Thus, $q_t(\mathbf{x}'|\mathbf{x})$ converges to a dirac-delta distribution $\delta_0(\mathbf{x}' - \mathbf{x})$. Thus, the marginal $q_t(\mathbf{x}') = p_t(\mathbf{x}')$ and IPM$_G(p_t, q_t) = 0$. Also, the $\delta$ in Assumption 5.8 converges to 0.

## C. Detailed Description of the Datasets

Here, we present a detailed description of the various datasets used in our work.

### C.1. Binary Treatments

| Dataset | Covariates | Samples | Runs | Covariates Type | Treatment | Synthetic $\mathcal{Y}$? | Outcome |
|---|---|---|---|---|---|---|---|
| IHDP | 25 | 747 | 100 | Features of an infant | Intervention by specialist doctor | ✓ | Cognitive test scores |
| ACIC | 58 | 4802 | 30 | Demographic, clinical features, etc. of pregnant women | N/A | ✓ | Developmental disorders |
| Twins | 40 | 11,984 | 100 | Characteristics of same-sex twins both weighing less than 2kg | Heavier twin | ✗ | One-year infant mortality |

Table 10: This table provides information about the Binary datasets used in our work. All these datasets exhibit treatment selection bias, meaning that the observed treatments are influenced by the covariates. The 'Runs' column indicates the number of experiments conducted for each dataset, from which the associated $p$-values are calculated. Notably, the ACIC dataset encompasses three distinct potential outcome functions (versions 2, 7, 26), each repeated for 10 seeds, resulting in a total of 30 runs. For the cases where $\mathcal{Y}$ is synthetic, the assumed potential outcome functions that generate the observed outcomes can at most be interpreted as modeling the original intended outcomes.

**IHDP.** The Infant Health and Development Program (IHDP) is a randomized controlled trial designed to evaluate the impact of physician home visits on the cognitive test performance of premature infants. The dataset exhibits selection bias, as non-random subsets of treated individuals are deliberately removed from the training dataset. Since we have observed outcomes for only one treatment, to render the dataset suitable for causal inference, we generate both observed and counterfactual outcomes using a synthetic outcome generation function based on the original covariates considering both treatments. The IHDP dataset comprises 747 subjects and includes 25 variables. While the original dataset discussed in (Shalit et al., 2017) had 1000 versions, a smaller version of the dataset with 100 versions is used in our work, aligning with the CATENets benchmark. Each version varies in terms of the complexity of the assumed outcome generation function, treatment effect heterogeneity, etc. As outlined in (Curth et al., 2021), reporting the standard deviation of performance across the 100 different seeds is inappropriate and therefore we calculate $p$-values through paired t-tests between our method (PairNet) and other baseline methods such that PairNet serves as the baseline for all experiments. Specifically, we accept the hypothesis that PairNet is superior to the baseline if the resulting $p$-value is less than 0.05.

**ACIC.** The Atlantic Causal The Atlantic Causal Inference Conference competition dataset (2016)[3] contains a total of 77 datasets. The covariates in all these datasets are the same and contain 58 features obtained from a real study called the Collaborative Perinatal Project. Each dataset involves simulating binary treatment assignments and continuous outcome variables. The datasets exhibit variations in several aspects, including the complexity of the treatment assignment mechanism, treatment effect heterogeneity, the ratio of treated to control observations, overlap between treatment and control groups, the dimensionality of confounder space, and the magnitude of the treatment effect. All datasets share common characteristics such as independent and identically distributed observations conditional on covariates, adherence to the ignorability assumption (selection on observables with all confounders measured and no hidden bias), and the presence of non-true confounding covariates. Of these 77 datasets, we opted to work with a subset of three datasets, specifically versions 2, 7, and 26, aligning with the CATENets benchmark. These three settings present non-linear covariate-to-outcome relationships and showcase maximum variability in terms of treatment effect heterogeneity. Notably, version 2 exhibits no heterogeneity, i.e. the treatment effect remains constant across all individuals. However, accurately estimating the outcome differences even for this version proves challenging as algorithms find it difficult to overcome the inherent noise observed in potential outcome realizations in the dataset. Specifically,

[3] https://jenniferhill7.wixsite.com/acic-2016/competition

in PairNet  we found that coming up with good pairs is important as we will explain in detail in the Table 18 in our main paper. The other two dataset versions show medium and high heterogeneity in terms of the treatment effects across the individuals.

**Twins.** The Twins dataset (Louizos et al., 2017) stands out as the sole dataset with actual observed outcomes. This study operates on the premise that the two twins within each pair share equivalence across all covariates, differing solely in terms of their treatment assignments. This unique characteristic allows the dataset to be employed as is for causal inference tasks. The dataset encompasses a total of 11,984 pairs of twins and focuses on one-year mortality as a function of birth weight, serving as the underlying treatment variable. To ensure the covariate equivalence, the study exclusively includes same-sex twins with birth weights below 2kg. In total, the dataset incorporates 39 relevant covariates. The dataset's outcomes are binary and exhibit a class imbalance in the observed outcomes. Thankfully, the mortality rates are low and stand at 16.1% for the treated group and 17.7% for the untreated group. Consequently, observing twin pairs with opposite outcomes in the dataset is a rare occurrence. In our experiments, we allocate 50% of the dataset for testing purposes. To introduce imbalance in the treated vs. control examples in the training dataset, we sample the treated group for each twin pair using probabilities from the set $\{0.1, 0.25, 0.5, 0.75, 0.9\}$. For each of these probabilities, we explore the sample efficiency of various methods by varying the number of training examples across the range of $\{500, 1000, 4000, 5700\}$. These experiment settings closely match that of the CATENets benchmark.

### C.2. Continuous Treatments

In this section, we provide a more detailed description of the five continuous benchmark datasets used in our work.

| Dataset | Covariates | Samples | Runs | Covariates Type | Dosage | Outcome |
|---|---|---|---|---|---|---|
| TCGA (0–2) (Bica et al., 2020) | 4000 | 9659 | 10 | Gene expressions of cancer patients | dosage of the drug | cancer recurrence |
| IHDP (Nie et al., 2021) | 25 | 747 | 50 | features of an infant | Amount of intervention by a specialist doctor | cognitive test scores |
| News (Dua & Graff, 2017) | 2858 | 3000 | 20 | Bag-of-words from news articles | time spent reading the news article | user-satisfaction |

Table 11: This table shows the information on the Continuous datasets where the treatment assignment and the corresponding responses are synthesized. The assumed potential outcome functions that generate the observed $y$ can at most be interpreted as modeling the original intended outcomes.

**TCGA (0–2) (Bica et al., 2020).** The TCGA dataset, sourced from The Cancer Genome Atlas project, contains information on various cancer types for a total of 9659 individuals. Each individual is characterized by 4000 dimensions of gene expression covariates. These covariates have been log-normalized and subsequently normalized to achieve unit variance. The treatment variable signifies the dosage of the drug administered to the patient, while the synthetic response models the risk of cancer recurrence. In our experiments, we make use of three versions of the TCGA dataset introduced in (Bica et al., 2020), denoted as TCGA(0), TCGA(1), and TCGA(2).

**IHDP.** The IHDP dataset was originally collected as part of the Infant Health Development Program for binary treatment effect estimation. Treatments in this dataset were assigned through a randomized experiment and it comprises 747 subjects with 25 covariates. For the continuous treatment effect problem, the dataset was adapted in (Nie et al., 2021) by assigning synthetic treatments and targets.

**News (Dua & Graff, 2017).** The News dataset was initially designed for binary treatment effect estimation and was adapted in (Bica et al., 2020) to accommodate continuous treatments and targets. In this dataset, the treatment variable represents the time spent by a user reading a news article, while the synthetic response aims to mimic user satisfaction. The dataset comprises 3000 randomly selected articles from the New York Times, with 2858 bag-of-words covariates.

For each of these datasets, we generate multiple versions using different random seeds, as per previous research.

### C.3. Continuous Dosage and Response Generation Functions

We adopt the dataset generation procedures from previous methods and **quote** a detailed explanation of the process here for completeness, following (Schwab et al., 2020; Nie et al., 2021; Bica et al., 2020).

**IHDP.** For treatments in the interval $[0,1]$, we generate responses as follows:

$$\tilde{t}|\mathbf{x}=\frac{2x_1}{1+x_2}+\frac{2\max(x_3,x_5,x_6)}{0.2+\min(x_3,x_5,x_6)}+2\tanh\left(\frac{5\sum_{i\in S_{\text{dis},2}}(x_i-c_2)}{|S_{\text{dis},2}|}-4+\mathcal{N}(0,0.25)\right), \tag{6}$$

$$t=(1+\exp(-\tilde{t}))^{-1}, \tag{7}$$

$$y|\mathbf{x},t=\frac{\sin(3\pi t)}{1.2-t}\cdot\tanh\left(\frac{5\sum_{i\in S_{\text{dis},1}}(x_i-c_1)}{|S_{\text{dis},1}|}\right)+\frac{\exp(0.2(x_1-x_6))}{0.5+5\min(x_2,x_3,x_5)}+\mathcal{N}(0,0.25), \tag{8}$$

where $S_{\text{con}}=\{1,2,3,5,6\}$ is the index set of continuous covariates, $S_{\text{dis},1}=\{4,7,8,9,10,11,12,13,14,15\}$, $S_{\text{dis},2}=\{16,17,18,19,20,21,22,23,24,25\}$, and $S_{\text{dis},1}\bigcup S_{\text{dis},2}=[25]-S_{\text{con}}$. Further, $c_1=\mathbb{E}\left[\frac{\sum_{i\in S_{\text{dis},1}}x_i}{|S_{\text{dis},1}|}\right]$ and $c_2=\mathbb{E}\left[\frac{\sum_{i\in S_{\text{dis},2}}x_i}{|S_{\text{dis},2}|}\right]$.

**News.** For the News dataset, we first generate $v_1',v_2',v_3'$ from $\mathcal{N}(0,1)$ and set $v_i=\frac{v_i'}{\|v_i'\|}$. The treatment and response generation is as follows:

$$t|\mathbf{x}=\text{Beta}\left(2,\left\|\frac{v_3^\top}{2v_2^\top\mathbf{x}}\right\|\right), \tag{9}$$

$$y'|\mathbf{x},t=\exp\left(\frac{v_2^\top\mathbf{x}}{v_3^\top\mathbf{x}}-0.3\right), \tag{10}$$

$$y|\mathbf{x},t=2(\max(-2,\min(2,y'))+20v_1^\top\mathbf{x})(4(t-0.5)^2+\sin\left(\frac{\pi}{2}t\right)+\mathcal{N}(0,0.5). \tag{11}$$

**TCGA(0-2).** For TCGA(0-2), we first generate $v_1',v_2',v_3'$ from $\mathcal{N}(0,1)$ and set $v_i=\frac{v_i'}{\|v_i'\|}$. We then add noise $\epsilon\sim\mathcal{N}(0,0.2)$. The dosage $d|\mathbf{x},t$ follows a Beta distribution with parameter $\alpha$ (default as 2) representing the dosage selection bias. We calculate $t_t=\frac{\alpha-1}{d^*}+2-\alpha$, with $d^*$ as the optimal dosage for that treatment.

For TCGA(0), we generate $y$ and $d^*$ as follows:

$$y|\mathbf{x},d=10(v_1^\top\mathbf{x}+12dv_3^\top\mathbf{x}-12d^2v_3^\top\mathbf{x}), \tag{12}$$

$$d^*=\frac{v_2^\top\mathbf{x}}{2v_3^\top\mathbf{x}}. \tag{13}$$

For TCGA(1), we generate $y$ and $d^*$ as follows: $y|\mathbf{x},d=10((v_1)^\top\mathbf{x}+\sin(\pi(\frac{v_2^\top\mathbf{x}}{v_3^\top\mathbf{x}}d))),d^*=\frac{v_3^\top\mathbf{x}}{2v_2^\top\mathbf{x}}$.

For TCGA(2), $y|\mathbf{x},d=10(v_1^\top\mathbf{x}+12d(d-0.75\frac{v_2^\top\mathbf{x}}{v_3^\top\mathbf{x}})^2),d^*=0.25\frac{v_2^\top\mathbf{x}}{v_3^\top\mathbf{x}}$ if $\frac{v_2^\top\mathbf{x}}{v_3^\top\mathbf{x}}\geq 1$, else 1.

## D. Network Architecture

**Binary Datasets:** PairNet is a representation learning-based method. It uses the same architecture as other representation learning-based baselines. The model comprises a Representation Network ($\phi$) with three layers, each consisting of 200 units and ELU (Exponential Linear units) activation functions. Additionally, it includes two $\mu$ Networks, $\mu_0$ (for T=0) and $\mu_1$ (for T=1). These networks follow a similar structure with two layers, featuring 100 units each and ELU activation functions. PairNet estimates the treatment effect $\tau$ as the difference between the two predicted potential outcomes.

| **Network Architecture for Binary Datasets** | |
|---|---|
| **Input Data** | |
| **Representation Network** ($\phi$) | |
| Layer 1: | 200 Units, ELU Activation |
| Layer 2: | 200 Units, ELU Activation |
| Embedding Layer: | 200 Units, ELU Activation |
| $\mu_0$ **Network (T=0)** | |
| Layer 1: | 100 Units, ELU Activation |
| Layer 2: | 100 Units, ELU Activation |
| Output Layer: | 1 Unit |
| $\mu_1$ **Network (T=1)** | |
| Layer 1: | 100 Units, ELU Activation |
| Layer 2: | 100 Units, ELU Activation |
| Output Layer: | 1 Unit |

**Continuous Datasets.** Here, we consider two architectures namely DRNet, VCNet

**DRNet:** We borrow the architecture of the dose-response network (Schwab et al., 2020) which is an extension of the binary treatment effect estimation models to the continuous case. Instead of two heads as in the case of binary treatment, we split the continuous treatments $T \in [0,1]$ into $N=5$ uniformly spaced bins and assign a separate output head $\mu_k$ for the $k^{th}$ bin, $k \in \{0,1,...,4\}$. Additionally, for every sample, the treatment $T=t$ is first normalized by subtracting the magnitude of the lower bin edge $\frac{k}{N}$ and then this scalar is concatenated with the input to each linear layer for the corresponding output head $\mu_k$. For PairNet we estimate the treatment effect $\tau$ for any two treatments $t,t'$ as the difference between predicted potential outcomes of the corresponding $\mu$ heads.

**VCNet:** The varying coefficient network (Nie et al., 2021) has the same representation network $\phi$ followed by a single output head $\mu(t)$ where the weights are parameterised by the treatment $t$. Let us denote the weights of the $\mu$ head for a treatment $t$ as $\theta(t) \in \mathbb{R}^d$. Each weight $\theta_i(t)$ is obtained as a smooth function of the treatment $t$. In particular, we apply spline basis expansion on $t$ using $N=5$ spline basis functions $\{\alpha_i\}_{i=1}^5$ and then obtain $\theta_i(t)$ for all $i \in [d]$ as $\sum_{k=1}^5 a_{ik}\alpha_k(t)$. So, to learn $\mu$ means to learn the coefficients of the linear combination $\{a_{ik}\}$ while keeping the basis functions $\alpha_i(.)$ fixed. For $\{\alpha_i\}_{i=1}^5$, we use the truncated polynomial basis used by (Nie et al., 2021) with degree 2 and knots at $\{\frac{1}{3},\frac{2}{3}\}$. This results in a basis with the functions $\{1,t,t^2,(\textsc{ReLU}(t-\frac{1}{3}))^2,(\textsc{ReLU}(t-\frac{2}{3}))^2\}$.

In every forward pass, we first initialize the weights of the $\mu(t)$ head for the corresponding treatment $t$ and then pass the representation $\phi(x)$ as input to it. The smooth variation of weights with $t$ allows the network to learn a smooth potential outcome function. PairNet estimates the treatment effect $\tau$ for treatments $t,t'$ as $\mu(t')(\phi(x)) - \mu(t)(\phi(x))$

## E. PairNet loss for Twins Dataset with Binary outcome

When dealing with binary outcomes $\mathcal{Y} \in \{0,1\}$, modeling the difference of observed outcomes for two individuals transforms the problem into a three-way classification task, where the difference labels can take values across $\{-1,0,+1\}$. To solve this three-way classification task, PairNet however, uses the same architecture as used for other datasets, and converts the estimated potential outcomes into three-way classification logits as shown below. Let

$$
\begin{aligned}
\hat{y_0} &= \mu(\phi(\mathbf{x}),t=0) = P(y=0|\mathbf{x},t=0) \\
\hat{y_1} &= \mu(\phi(\mathbf{x}),t=1) = P(y=0|\mathbf{x},t=1) \\
\hat{y_0'} &= \mu(\phi(\mathbf{x}'),t=0) = P(y'=0|\mathbf{x}',t'=0) \\
\hat{y_1'} &= \mu(\phi(\mathbf{x}'),t=1) = P(y'=0|\mathbf{x}',t'=1)
\end{aligned} \tag{14}
$$

where we assume a pair of examples $(\mathbf{x},t,y,\mathbf{x}',t',y')$. Recall that by virtue of pairs creation, we have $t \neq t'$. Having estimated all possible potential outcomes for the pairs, we compute the three-way logits as follows:

$$
\begin{aligned}
\text{logits}[-1] &= \left(\hat{y_0}\cdot(1-\hat{y_1'})\right)\cdot(1-t) + \left(\hat{y_1}\cdot(1-\hat{y_0'})\right)\cdot t \quad \text{(for the case when } y=0, y'=1) \\
\text{logits}[\ 0] &= \left(\hat{y_0}\cdot\hat{y_1'}\right)\cdot(1-t) + \left(\hat{y_1}\cdot\hat{y_0'}\right)\cdot t + \left((1-\hat{y_0})\cdot(1-\hat{y_1'})\right)\cdot(1-t) + \left((1-\hat{y_1})\cdot(1-\hat{y_0'})\right)\cdot t \quad \text{(for the case } y=y') \\
\text{logits}[+1] &= \left((1-\hat{y_0})\cdot\hat{y_1'}\right)\cdot(1-t) + \left((1-\hat{y_1})\cdot\hat{y_0'}\right)\cdot t \quad \text{(for the case when } y=1, y'=0)
\end{aligned} \tag{15}
$$

We finally impose a `cross-entropy` loss using the difference label $y-y'$ as the target.

## F. Computational Infrastructure and Default Hyper-parameters

Our experiments were conducted on a DGX machine equipped with an NVIDIA A100 GPU card, with 80 GB of GPU memory. The DGX machine is powered by an AMD EPYC 7742 64-Core Processor with 256 CPUs, featuring 64 cores per CPU. Our codebase was entirely developed using JAX (Bradbury et al., 2018), a functional programming-based deep learning library that extends CUDA support for GPU acceleration.

For a fair comparison, we adopt the hyperparameters used in the CATENets benchmark (Curth et al., 2021) as is except for the weight associated with the $L_2$ penalty. In each epoch of training, we sample mini-batches of 100 examples (along with their respective pairs for PairNet) and impose losses on them. We use Adam optimizer with a learning rate set to 1e-4. Training proceeds for a maximum of 1000 epochs, while we perform early stopping based on a 30% validation set and a patience level of 10. In the case of binary datasets, we use stratified sampling on the treatments to obtain the validation split, while for continuous datasets, we use random sampling.

## G. On the impact of $L_2$ penalty

In this experiment, we analyze the performance of PairNet and baselines under different $L_2$ penalties on the $\phi$ parameters. In our primary results, which are presented in Table 1, we applied a high $L_2$ penalty scale of 1 to both PairNet and the baselines. Here we present the results when these methods are trained using CATENets default value of 1e-4.

For better exposition, we compare models trained with the default $L_2$ penalty (1e-4) to those trained with a strong penalty (1). To assess the statistical significance of these performance differences, we conduct $p$-tests using models from the main table 1 as baselines. For example, consider the value “-0.37 (0.01)” in the first cell of table 12. It indicates that training the TLearner on IHDP with a $1e-4$ $L_2$ penalty leads to an average error increase of 0.37 across various IHDP seeds, with a $p$-value of 0.01 showing the significance of this error increase. Throughout the table, a $p$-value below 0.05 suggests that a $L_2$ penalty of 1 is beneficial for the model's performance, while a $p$-value above 0.95 indicates that the strong penalty may hinder performance. In summary, we draw the following conclusions from Table 12:

1. Negative values for many methods indicate that a stronger penalty is beneficial for these methods.
2. Additionally, several $p$-values fall below 0.05, indicating statistically significant performance improvements resulting from the stronger $L_2$ penalty.
3. RLearner, which employs Robinson decomposition to directly model CATE, is the most affected by the strong $L_2$ penalty.
4. When comparing results across Tables 1 and 12, our results show that PairNet trained with an $L_2$ penalty of 1 outperforms all baselines trained on either value of $L_2$ penalty.

| | IHDP | | ACIC | | Twins | |
|---|---|---|---|---|---|---|
| | PEHE in | PEHE out | PEHE in | PEHE out | PEHE in | PEHE out |
| TLearner (Künzel et al., 2019) | -0.37 (0.01) | -0.42 (0.06) | -0.58 (0.21) | -0.95 (0.22) | -0.02 (0.00) | -0.03 (0.00) |
| RLearner (Nie & Wager, 2021) | +0.87 (0.99) | +0.75 (0.96) | +0.87 (0.95) | +0.15 (0.62) | -0.01 (0.00) | -0.02 (0.00) |
| DRLearner (Kennedy, 2020) | -0.39 (0.01) | -0.42 (0.06) | +0.22 (0.64) | -0.03 (0.48) | -0.01 (0.00) | -0.02 (0.00) |
| XLearner (Künzel et al., 2019) | -0.24 (0.21) | -0.29 (0.22) | -0.31 (0.29) | -0.75 (0.16) | -0.01 (0.00) | -0.01 (0.00) |
| TARNet (Künzel et al., 2019) | -0.52 (0.00) | -0.67 (0.00) | -0.30 (0.25) | -0.77 (0.05) | -0.01 (0.01) | -0.01 (0.00) |
| CFRNet (Shalit et al., 2017) | -0.59 (0.00) | -0.68 (0.00) | +0.01 (0.51) | -0.33 (0.30) | 0.00 (0.73) | +0.00 (0.84) |
| DragonNet (Shi et al., 2019) | -0.51 (0.00) | -0.66 (0.00) | -0.28 (0.26) | -0.74 (0.06) | -0.01 (0.00) | -0.01 (0.00) |
| FlexTENet (Curth & van der Schaar, 2021) | -0.21 (0.03) | -0.25 (0.10) | +0.67 (0.96) | +1.42 (0.99) | +0.04 (1.00) | +0.05 (1.00) |
| PairNet | -0.98 (0.00) | -1.08 (0.00) | -0.48 (0.09) | -0.89 (0.01) | -0.00 (0.50) | -0.00 (0.49) |

Table 12: Differences in Performance with Strong $L_2$ Penalty: We examine the performance differences between models trained under the default $L_2$ penalty setting (1e-4) and those trained with a strong $L_2$ penalty of 1 on the $\phi$ parameters. This table presents the mean differences and corresponding $p$-values in brackets for a one-sided paired t-test conducted using the methods from Table 1 as the baseline. The table illustrates the discrepancy in error compared to our primary results where negative values indicate that the strong $L_2$ penalty outperforms its counterpart. The direct methods are highlighted in green and representation learning-based methods are highlighted in yellow.

| | IHDP | | News | | TCGA-0 | |
|---|---|---|---|---|---|---|
| | PEHE in | PEHE out | PEHE in | PEHE out | PEHE in | PEHE out |
| DRNet (Schwab et al., 2020) | -1.38 (0.00) | -1.43 (0.00) | 0.13 (1.00) | -0.01 (0.29) | 0.06 (0.94) | 0.05 (0.91) |
| PairNet (DRNet) | -1.33 (0.00) | -1.91 (0.00) | -0.07 (0.00) | -0.27 (0.00) | -0.05 (0.03) | -0.07 (0.00) |
| VCNet (Nie et al., 2021) | 0.44 (1.00) | 0.17 (0.96) | -0.00 (0.32) | -0.02 (0.04) | -0.01 (0.35) | -0.01 (0.32) |
| PairNet (VCNet) | 0.34 (1.00) | 0.12 (0.83) | 0.01 (0.88) | -0.01 (0.09) | -0.00 (0.43) | -0.00 (0.42) |

Table 13: Performance of PairNet and baselines on DRNet, VCNet assessed using PEHE error on continuous datasets. We compute the difference in error for two different values of $L_2$ penalty, 1 and $10^{-4}$. We report differences in mean error and $p$-values in brackets for a one-sided paired t-test conducted with $L_2 = 1$ as the baseline for each model. We see that all the methods benefit significantly from regularizing the $\phi$ parameters.

## H. Results using Shallow Model Architecture

We performed experiments using the shallow architecture in (Curth et al., 2021). Our main paper featured three layers, each with 200 neurons, in $\phi$, and two layers, each with 200 neurons, in $\mu$. Here, we conducted an experiment using one layer with 200 neurons in $\phi$ and one layer with 100 neurons each in the $\mu$ heads. We show the PEHE-out for the five binary datasets in Table 14. With this smaller model, PairNet outperforms the baselines by a greater margin than what was reported in our main paper in Table 1, as indicated by the significance of many p-values in the above table.

| Method | IHDP | ACIC2 | ACIC7 | ACIC26 | Twins |
|---|---|---|---|---|---|
| TNet | 2.04 (0.02) | 4.14 (0.01) | 3.99 (0.1) | 3.95 (0.06) | 0.32 (0.00) |
| TARNet | 1.16 (0.2) | 1.78 (0.24) | 3.88 (0.12) | 3.75 (0.05) | 0.33 (0.00) |
| CFRNet | 1.34 (0.12) | 3.59 (0.2) | 4.56 (0.02) | 4.49 (0.02) | 0.33 (0.00) |
| RNet | 3.0 (0.00) | 1.18 (0.13) | 5.44 (0.00) | 4.91 (0.05) | 0.32 (0.19) |
| XNet | 2.18 (0.03) | 3.15 (0.07) | 3.88 (0.11) | 3.55 (0.08) | 0.32 (0.02) |
| FlexTENet | 1.35 (0.1) | 5.59 (0.02) | 4.54 (0.03) | 6.4 (0.01) | 0.36 (0.00) |
| DRNet | 1.38 (0.1) | 4.77 (0.11) | 4.51 (0.04) | 3.99 (0.02) | 0.32 (0.10) |
| DragonNet | 1.15 (0.2) | 1.74 (0.25) | 3.89 (0.12) | 3.78 (0.04) | 0.33 (0.00) |
| IPW | 1.12 (0.23) | 1.69 (0.24) | 3.43 (0.3) | 3.43 (0.07) | 0.33 (0.00) |
| NearNeighbor | 1.6 (0.08) | 1.2 (0.14) | 4.3 (0.03) | 4.3 (0.00) | 0.32 (0.00) |
| PairNet | 0.76 (0.00) | 0.7 (0.00) | 3.05 (0.00) | 2.36 (0.00) | 0.32 (0.00) |

Table 14: Performance comparison on shallow model architecture which features one layer with 200 neurons in $\phi$ and one layer with 100 neurons each in the $\mu$ heads. We show the PEHE-out values with $p$-values in brackets. We observe that the performance gains of using pair loss is much more evident on shallow architectures than what was observed on deep architecture shown in Table 1.

## I. Results on Individual ACIC Versions

The main Table 1 featured aggregated results for the ACIC datasets. We compare the performance of PairNet with other baselines on each version of the ACIC dataset in Table 15. We see that except on ACIC2, PairNet achieves much better performance on the remaining datasets.

| Method | ACIC2 | ACIC7 | ACIC26 |
|---|---|---|---|
| TNet | 3.39(0.01) | 3.50(0.1) | 6.00(0.14) |
| TARNet | 1.15(0.68) | 3.49(0.09) | 3.5(0.22) |
| CFRNet | 2.12(0.36) | 4.12(0.01) | 4.12(0.06) |
| RNet | 1.26(0.71) | 5.30(0.00) | 5.25(0.01) |
| XNet | 1.80(0.39) | 3.51(0.07) | 4.60(0.15) |
| FlexTENet | 4.89(0.00) | 4.62(0.00) | 6.60(0.00) |
| DRNet | 2.50-(0.28) | 3.85(0.03) | 3.65(0.16) |
| DragonNet | 1.15(0.68) | 3.49(0.09) | 3.52(0.21) |
| IPW | 1.35(0.59) | 2.85(0.53) | 3.5(0.24) |
| NearNeighbor | 1.24(0.72) | 4.09(0.00) | 4.10(0.06) |
| PairNet | 1.56(0.00) | 2.89(0.00) | 2.95(0.00) |

Table 15: Performance comparison on individual ACIC seeds.

## J. Results including PEHE in for continuous datasets

We present the results for PEHE in errors in table 16 for completeness. We observe that both PEHE in and PEHE out errors exhibit similar trends across the datasets.

| | IHDP | | News | | TCGA-0 | | TCGA-1 | | TCGA-2 | |
|---|---|---|---|---|---|---|---|---|---|---|
| | PEHE in | PEHE out | PEHE in | PEHE out | PEHE in | PEHE out | PEHE in | PEHE out | PEHE in | PEHE out |
| DRNet (Schwab et al., 2020) | 2.29 (0.00) | 2.45 (0.00) | 1.42 (0.00) | 1.42 (0.00) | 0.34 (0.01) | 0.34 (0.01) | 0.26 (0.07) | 0.26 (0.08) | 0.24 (0.02) | 0.24 (0.02) |
| PairNet (DRNet) | 2.03(0.00) | 2.27(0.00) | 1.21(0.00) | 1.32(0.00) | 0.26(0.00) | 0.26(0.00) | 0.22(0.00) | 0.22(0.00) | 0.21(0.00) | 0.21(0.00) |
| VCNet (Nie et al., 2021) | 1.59 (0.00) | 1.73 (0.02) | 1.13(1.00) | 1.24(1.00) | 0.25 (0.57) | 0.25 (0.57) | 0.22 (0.45) | 0.22 (0.45) | 0.21 (0.45) | 0.21 (0.46) |
| PairNet (VCNet) | 1.43(0.00) | 1.57(0.00) | 1.14 (0.00) | 1.26 (0.00) | 0.25 (0.00) | 0.25 (0.00) | 0.22 (0.00) | 0.22 (0.00) | 0.21 (0.00) | 0.21 (0.00) |

Table 16: Performance of PairNet losses on DRNet, VCNet assessed using PEHE error on continuous datasets. We report mean and $p$-values within brackets for a one-sided paired t-test conducted with PairNet as the baseline.

| | TCGA0 | | | TCGA2 | | |
|---|---|---|---|---|---|---|
| Trn. Size | 5410 | 1352 | 541 | 5410 | 1352 | 541 |
| Factual | 0.25 (0.51) | 0.28 (0.14) | 0.43 (0.02) | 0.45 (0.52) | 0.46 (0.44) | 0.58 (0.13) |
| PairNet | 0.26 (0.00) | 0.25 (0.00) | 0.27 (0.00) | 0.45 (0.00) | 0.46 (0.00) | 0.49 (0.00) |

Table 17: PEHE out error for VCNet

| | Random | PairNet's $\phi$ | $\psi$ |
|---|---|---|---|
| IHDP | 0.45 (0.42) | 0.46 (0.52) | 0.45 (0.00) |
| ACIC-2 | 1.26 (0.88) | 1.53 (0.84) | 0.69 (0.00) |
| IHDP (DRNet) | 2.28 (0.57) | 2.27 (0.60) | 2.29 (0.00) |
| IHDP (VCnet) | 1.57 (0.49) | 1.56 (0.51) | 1.57 (0.00) |

Table 18: PEHE out error for various schemes of deriving pairs both the binary and continuous treatment settings assessed across five seeds. Overall we observe that PairNet is not sensitive to the different choices, except in the ACIC-2 dataset when the $\psi$ performs the best.

| | PairNet | | | | |
|---|---|---|---|---|---|
| | A | B | A+B | Random | Factual |
| PEHE in | 0.98 (0.00) | 1.01 (0.48) | 1.00 (0.49) | 1.03 (0.47) | 1.03 (0.47) |
| PEHE out | 0.58 (0.00) | 0.61 (0.47) | 0.60 (0.48) | 0.64 (0.44) | 0.61 (0.47) |

Table 19: PEHE error on synthetic data under different pairing strategies for PairNet. Strategy A which relies on relevant covariates for pair construction outperforms other approaches highlighting the importance of a good embedding function.

## K. Impact of Embedding Function for Pairing

PairNet uses embeddings $\psi(\mathbf{x})$ to calculate distances when creating pairs. We evaluate the impact of different embedding functions on performance for both real and synthetic data and show the importance of a good distance measure. We consider three variants of embedding function: (a) Random: pairs are selected arbitrarily, (b) Factual $\psi$: uses embeddings from a pre-trained model, trained solely using the factual loss. (c) PairNet's $\phi$: pairs are generated at each epoch using distances computed on PairNet's representation network $\phi$, trained up to that epoch.

The results are shown in table 18. For IHDP, we see that PairNet exhibits resilience to different choices for the embedding function. Similar trends are seen for most other datasets (not shown here). However, for the second version of the the ACIC dataset, we notice that the use of $\psi$ leads to a significant reduction in error over random selection. This indicates that finding a good distance measure for samples can boost the performance of our algorithm but even for a poor measure we do not do worse than random selection in most cases.

## L. Correct Pairing Boosts PairNet

We further conduct a toy experiment to bring out the point that finding correct pairs boosts PairNet's performance while arbitrary pairing does not hurt beyond a factual model. We considered i.i.d. Gaussian covariates $\mathbf{x} \in \mathbb{R}^{10}$. We assumed $\mu_0$, $\mu_1$ to be 3-degree polynomials such that their difference $\tau$ is of degree-2. Suppose $\mu_0$,$\mu_1$ solely depends on the first 5 covariates, we test four distance measure variants for pair computation in PairNet: (1) A - computes distances using $\mathbf{x}[0:5]$ (2) B - uses irrelevant covariates $\mathbf{x}[5:10]$ (3) A+B uses all covariates (4) Random - arbitrary pairing

As expected, PairNet-A outperforms others, emphasizing the significance of nearby pairing and a robust distance measure. This observation aligns with PairNet's $\phi_{\text{fct}}$, extracting predictive features for $\mu$, analogous to $\mathbf{x}[0:5]$ here.

To avoid over-representation of certain examples while pairing, we also considered techniques like Optimal Transport, but they did not perform much better than the presented approaches.

## M. Details on the Code

We have uploaded the code with our supplementary material. The code is accompanied by a `README.MD` that specifies the installation instructions, and directions on how to run the code.

## N. Limitations

These are the main limitations of our work:

1. PairNet imposes losses only on the factual observed outcomes. Therefore, even for the cases where distant examples are paired, its performance is not expected to degrade beyond models trained solely on factual losses. We face the following challenges in finding good pairs:
   - There should be an adequate number of proximate pairs in the observational dataset for PairNet to find them.
   - The requirement for proximal pairs with opposite treatments in the training data is mitigated by the overlap assumption that is generally made for causal inference tasks. Overlap states that $P(t|X) > 0 \;\forall\, t \in T$. However, with finite observational datasets, this assumption may not hold for certain covariates, resulting in the pairing of distant ones.
2. Another limitation arises when opting to apply losses to more than one pair ($\text{num}_{z'} > 1$) for each observed sample. This results in a computational time increase, albeit only linearly proportional to $\text{num}_{z'}$. Furthermore, generating pairs may be time-consuming, especially when dealing with large datasets. Nevertheless, efficient techniques such as FAISS (Johnson et al., 2019) can be employed to perform the pairing efficiently.