Continuous Treatment Effect Estimation Using Gradient Interpolation and Kernel Smoothing

Lokesh Nagalapatti, Akshay Iyer, Abir De, Sunita Sarawagi

Indian Institute of Technology Bombay
nlokesh@cse.iitb.ac.in, akshaygiyer@gmail.com, abir@cse.iitb.ac.in, sunita@iitb.ac.in

Abstract

We address the Individualized continuous treatment effect (ICTE) estimation problem where we predict the effect of any continuous-valued treatment on an individual using observational data. The main challenge in this estimation task is the potential confounding of treatment assignment with an individual's covariates in the training data, whereas during inference ICTE requires prediction on independently sampled treatments. In contrast to prior work that relied on regularizers or unstable GAN training, we advocate the direct approach of augmenting training individuals with independently sampled treatments and inferred counterfactual outcomes. We infer counterfactual outcomes using a two-pronged strategy: a Gradient Interpolation for close-to-observed treatments, and a Gaussian Process based Kernel Smoothing which allows us to downweigh high variance inferences. We evaluate our method on five benchmarks and show that our method outperforms six state-of-the-art methods on the counterfactual estimation error. We analyze the superior performance of our method by showing that (1) our inferred counterfactual responses are more accurate, and (2) adding them to the training data reduces the distributional distance between the confounded training distribution and test distribution where treatment is independent of covariates. Our proposed method is model-agnostic and we show that it improves ICTE accuracy of several existing models. We release the code at: https://github.com/nlokeshiisc/GIKS_release.

1 Introduction

Many applications require the estimation of the effect of a continuous treatment variable on an individual's response. For example, in healthcare we need to estimate the effect of the dose of a drug on the recovery of a patient, in economics, we need to estimate the effect of a discount on the sales of a product, and in public policy, we need to estimate the effect of income on a person's longevity. In all these cases, observational data is available in abundance but controlled experiments to estimate these effects exactly either raise ethical issues or incur hidden costs. The main challenge in estimating treatment effects from observational data is that in the observed data each individual is associated with one treatment dose which may be correlated with observed covariates of the individual, but during deployment, we need to estimate outcomes on *all* treatment doses, thereby making dose *independent* of the individual in the test data distribution.

Several prior work have proposed to correct the above mismatch but most of these have focused on binary treatments. Broadly, most methods rely on a combination of these three strategies: (1) Learn shared representation of the feature with treatment specific outcome prediction (Shalit, Johansson, and Sontag 2017), (2) Impose regularizers to make feature representations distributionally independent of the treatment (Shalit, Johansson, and Sontag 2017), (3) Exploit the overlap assumption to impose instance-specific counterfactual losses in the learned feature space (Alaa and Van Der Schaar 2017; Zhang, Bellot, and Schaar 2020). Recently, a subset of these strategies have been extended to the case of continuous treatments where some have focused on extending the neural architecture to handle continuous treatments (Schwab et al. 2020; Nie et al. 2021; Zhang et al. 2022), and others extend the distributional regularizers (Nie et al. 2021; Bellot, Dhir, and Prando 2022). However, as we show in our experiment evaluation, such regularizers are not too effective in reducing *individual* continuous treatment effect estimation errors.

In this paper, we propose to directly minimize counterfactual loss for each individual by inferring outcomes at independently sampled new treatments. We infer outcomes using two types of smoothing strategies. First, by exploiting the differentiability of the response function to treatment, we infer the counterfactual response by gradient interpolation (Nasery et al. 2021). Second, by exploiting the property of overlap required for the identifiability of ICTE from observational data, we infer a counterfactual response by using a Gaussian process over feature kernels. We handle the potential unreliability of the inferred outcomes by downweighing examples based on the variance of the GP estimate. We show that individual-level counterfactual losses are significantly more effective in learning ICTE compared to distributional regularizers, particularly in a mini-batch setting. We attribute the reasons for the observed gains of our method to two factors: (1) the inferred outcomes from the data are more accurate than a baseline that is just trained on an observational dataset, and (2) the augmented data makes the training distribution closer to the test distribution.

We make the following contributions in this paper:

- We address the ICTE problem by directly minimizing loss on inferred counterfactual outcomes of new treatments applied to training instances. We infer counterfactual outcomes by (1) gradient interpolation justified by the differentiability of the response function to treatments, and (2) kernel smoothing based on the overlap assumption of covariates and treatment.

- We evaluate our method on five benchmark data and show that we consistently outperform six existing state-of-the-art methods on ICTE. We also demonstrate the application of our model in two medical settings.
- We explain the reasons for the observed gains by showing that our proximity-inferred outcomes are more accurate than the factual model, and the augmentation reduces the confounding between treatment and covariates.
- We show that our method is model-agnostic and provides gains on several existing model architectures.

2 Problem Formulation

We use random variables: $X \in \mathcal{X} \subset \mathbb{R}^{d_x}$ for the individual's covariates, $T \in \mathcal{T} \subset \mathbb{R}$ for treatments, and $Y(t) \in \mathbb{R}$ for potential outcomes or response when an individual is given treatment t. Our objective is to estimate the individual treatment effect $\mathbb{E}[Y(t)|\mathbf{x}]$, which represents the expected outcome when an individual $\mathbf{x} \in \mathcal{X}$ receives treatment t. In prior work, this term has also been referred to as the Average Dose-Response Function (ADRF), denoted as $\mu(\mathbf{x}, t)$. We adopt the Neyman–Rubin causal model (Pearl and Press 2000) and estimate ADRF using the Potential Outcomes Framework. The primary challenge is to learn $\mu(\mathbf{x}, t)$ from an observational dataset where each $\mathbf{x}$ is exposed to only one treatment dose, whose selection depends on $\mathbf{x}$ making the covariates correlated with the treatment.

The observational dataset D comprises N samples $\{(\mathbf{x}_i, t_i, y_i)\}_{i=1}^{N}$. Each $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^{d_x}$ denotes the covariate observed before an individual is exposed to a treatment; $t_i \in \mathcal{T} \subset \mathbb{R}$ is the value of continuous treatment T applied on the i-th instance and $y_i \in \mathbb{R}$ captures the outcome observed for $\mathbf{x}_i$ under the treatment dose t_i. We use $t_i^{\text{CF}} \neq t_i$ to denote any new treatment that is not observed in D.

Following prior work, to ensure the identifiability of ICTE from the observational dataset, we make the following assumptions:

A1 *Overlap of treatment:* which states that every individual has a non-zero probability of being assigned any treatment *i.e.*, $\Pr(t|\mathbf{x}) \in (0, 1)\ \forall \mathbf{x} \in \mathcal{X}, t \in \mathcal{T}$

A2 *identifiability of causal effect:* which states that the observed covariates $\mathcal{X}$ block all the backdoor paths between the treatments $\mathcal{T}$ and outcomes Y.

A3 *Differentiability of ADRF:* We assume that ADRF $\mu(\mathbf{x}, t)$ is differentiable w.r.t. treatment t.

Assumptions A1, and A2 are standard assumptions needed for causal inference and under them, we can claim that $\mathbb{E}[Y(t)|X] = \mathbb{E}[Y|t, X]$.

Following prior work (Nie et al. 2021; Shalit, Johansson, and Sontag 2017; Schwab et al. 2020), we model the ADRF μ using a composition of two neural networks as $\mu(\mathbf{x}, t) = \eta(\Phi(x), t)$, where $\Phi : \mathbb{R}^{d_x} \to \mathbb{R}^{d_e}$ embeds the covariates $\mathbf{x}$, and $\eta : \mathbb{R}^{d_e} \times \mathbb{R} \to \mathbb{R}$ predicts the average response given the embedding of an individual $\Phi(\mathbf{x})$, at a treatment dose t. Many recent models e.g. DRNet (Schwab et al. 2020), VCNet (Nie et al. 2021) follow this framework. Our default choice is to model the embedding network Φ using a simple feed-forward neural network and to make the η network sensitive to t using VCNet (Nie et al. 2021), a state-of-the-art network for continuous treatment effect inference. The parameters $\theta \in \mathbb{R}^{d_\theta}$ of the η network are obtained as predictions from another network $G_\psi : \mathbb{R} \to \mathbb{R}^{d_\theta}$. The G_ψ network uses spline bases to ensure a smooth variation of θ with t. Thus, we can express $\eta\big(\Phi(\mathbf{x}), t\big) = \eta\big(\Phi(\mathbf{x}); \theta = G_\psi(t)\big)$, and the only trainable parameters of η are the parameters of G_ψ *i.e.* the ψ.

The main challenge in learning the parameters of μ using observational data through standard likelihood training is the discrepancy between the training and test data distributions. The observational dataset D could confound treatment with $\mathbf{x}$, leading to dependence between the observed treatments and $\mathbf{x}$. Specifically, the training instances are drawn from $P(X, T) = P(T|X)P(X)$, where $T \not\perp\!\!\!\perp X$. On the other hand, during inference, we aim to estimate ADRF for an individual under arbitrarily assigned treatments, implying that $T \perp\!\!\!\perp X$. This corresponds to the test instances being drawn from $P^{\text{CF}}(X, T) = P(X)P(T)$. In the following, we show how we address this issue of disparity.

3 GIKS: Our Proposed Approach

Our main idea is to bridge the gap between the train and test distributions with inferred counterfactuals from auxiliary layers that harness data proximity. We start with a base model with Φ and η trained with factual loss on the training data: $\min_{\Phi,\eta} \sum_{i=1}^{N} \ell\big(\eta(\Phi(\mathbf{x}_i), t_i), y_i\big)$. Note that our losses are model agnostic, they can be integrated on top of several base architectures (*c.f.* Section 5.2). Thereafter, for each instance $(\mathbf{x}_i, t_i, y_i)$ in the training set, we sample new treatments t_i^{CF} from $P(T)$ and since there is no supervision for $y_i(t_i^{\text{CF}})$, we infer pseudo-targets $\widehat{y}_i(t_i^{\text{CF}})$ by leveraging the proximity in the $\Phi(X)$ and T space to other training examples in two ways: First, we use the ADRF differentiability assumption (A4) to predict responses for new treatments that are within a small distance δ of the observed treatment using Taylor's expansion of η, and impose a loss, which we call the GI loss $\mathcal{L}_{\text{GI}}$. Second, for treatments with a larger distance $|t_i^{\text{CF}} - t_i| > \delta$, we rely on the overlap assumption (A1) and employ kernel smoothing in the embedding space $\Phi(\mathbf{x})$ over samples $\mathbf{x}_j$ in D, whose observed treatments are close to t_i^{CF}. This gives us an inferred $\widehat{y}_i(t_i^{\text{CF}})$ and variance $\widehat{\sigma}^2(\widehat{y}(t_i^{\text{CF}}))$. We use these to impose a confidence-weighted loss which we call $\mathcal{L}_{\text{KS}}$. We elaborate on these two losses:

3.1 Gradient Interpolated Inferred Counterfactual Outcomes

Since, we assume that the learned ADRF $\mu(\mathbf{x}_i, t_i) = \eta(\Phi(\mathbf{x}_i), t_i)$ is differentiable w.r.t. t, for any new t_i^{CF} that lies close to t_i, we can infer its response using a first-order Taylor series expansion of μ around t_i^{CF}, and use these to impose instance-specific counterfactual loss as follows:

$$\min_{\Phi,\eta} \mathcal{L}_{\text{GI}} = \sum_{i=1}^{N} \mathcal{L}(\eta(\Phi(\mathbf{x}_i), t_i^{\text{CF}}), \widehat{y}_i(t_i^{\text{CF}})) \tag{1}$$

$$\text{where } \widehat{y}_i(t_i^{\text{CF}}) = y_i - (t_i - t_i^{\text{CF}}) \frac{\partial \eta(\Phi(\mathbf{x}_i), t)}{\partial t} \tag{2}$$

Despite the spline parameterization of the VCNet, we show in Figure 3, and Table 4 that $\eta(\Phi(\mathbf{x}), t)$ is not smooth enough, and the above GI loss helps. Note that the above loss is different from the gradient penalty used in other methods (Alvarez Melis and Jaakkola 2018; Arjovsky et al. 2019), where the gradient norm $\|\frac{\partial \eta}{\partial t}\|$ is used as a regularizer. Earlier work (Nasery et al. 2021) has shown that such GI induced losses are more effective in increasing the smoothness of a deep network on a continuous input, rather than the proposals by Alvarez Melis and Jaakkola (2018) and Arjovsky et al. (2019).

3.2 Kernel Smoothed Inferred Counterfactual Outcomes

To infer counterfactual responses for new treatments that are distant from the observed ones, we leverage the first assumption of Overlap: $P(t^{\text{CF}}|\mathbf{x}) > 0$ for all $\mathbf{x}$. On finite training data D, this implies that we need to rely on neighbors in D from the $\Phi(X)$ and T space to infer counterfactual outcomes. However, two challenges arise: (1) combining proximity in both the high-dimensional $\Phi(X)$ and low-dimensional T space, and (2) unreliability of responses inferred from sparse neighborhoods. We describe next how we handle these challenges via a Gaussian Process based estimator:

Gaussian Process for Estimating Counterfactual Response Suppose we wish to infer $\widehat{y}_i(t_i^{\text{CF}})$ for an observation $(\mathbf{x}_i, y_i, t_i)$. One option is to design a joint kernel over $\Phi(X)$ and T, for example, the product kernel(Bellot, Dhir, and Prando 2022) or the Neural Tangent Kernel (NTK) (Jacot, Gabriel, and Hongler 2018) derived from the Φ, η network. However, we found the following two-stage approach with a few learned parameters to be more effective.

First we account for proximity in T space, by collecting the instances $(\mathbf{x}_j, t_j, y_j)$, whose observed treatments t_j are close to t_i^{CF}, *i.e.*, $|t_i^{\text{CF}} - t_j| \leq \epsilon_{\text{GP}}$, and define a nearest neighbor dataset $D_{\text{NN}}(t_i^{\text{CF}})$:

$$D_{\text{NN}}(t_i^{\text{CF}}) = \{(\mathbf{x}_j, t_j, y_j) \in D| \ |t_i^{\text{CF}} - t_j| \leq \epsilon_{\text{GP}}\}, \tag{3}$$

$$\boldsymbol{X}_{\text{NN}}, \boldsymbol{y}_{\text{NN}} = [\mathbf{x}_j \in D_{\text{NN}}(t_i^{\text{CF}})], [y_j \in D_{\text{NN}}(t_i^{\text{CF}})]. \tag{4}$$

Then, to account for proximity in Φ space, we fit a Gaussian Process (GP) using $D_{\text{NN}}(t_i^{\text{CF}})$ as an inducing set (Titsias 2009) to infer counterfactual responses. Specifically, we model $y_i(t_i^{\text{CF}})$ as:

$$y_i(t_i^{\text{CF}}) = f(\Phi(\mathbf{x}_i)) + \epsilon, \text{ where } \epsilon \sim \mathcal{N}(0, \sigma^2) \quad \text{with,}$$
$$f(\Phi(\mathbf{x}_i)) \sim GP(0, K(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i')))$$

Then we estimate $\widehat{y}_i(t_i^{\text{CF}})$, the mean of the posterior as:

$$\widehat{y}_i(t_i^{\text{CF}}) = K(\Phi(\mathbf{x}_i), \Phi(\boldsymbol{X}_{\text{NN}})) V_{\text{NN}}^{-1} \boldsymbol{y}_{\text{NN}}. \tag{5}$$

where $V_{\text{NN}} = [\sigma^2 \mathbb{I} + K(\Phi(\boldsymbol{X}_{\text{NN}}), \Phi(\boldsymbol{X}_{\text{NN}}))]$. Further variance of the estimate $\widehat{\sigma}^2[y_i(t_i^{\text{CF}})]$ is given by:

$$\begin{aligned} &K(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_i)) \\ &\quad - K(\Phi(\mathbf{x}_i), \Phi(\boldsymbol{X}_{\text{NN}})) V_{\text{NN}}^{-1} K(\Phi(\mathbf{x}_i), \Phi(\boldsymbol{X}_{\text{NN}})). \end{aligned} \tag{6}$$

We leverage the GP variance to down-weigh unreliable outcomes and mitigate their impact on learned parameters. In particular, a large variance indicates a lack of nearby neighbors in $D_{\text{NN}}(t_i^{\text{CF}})$ for instance $\mathbf{x}_i$. Thus, in a mini-batch setting, we first sample $t_i^{\text{CF}} \sim P(T)$ for each individual in the batch, and then obtain mean and variance estimates using the GP for instances where $|t_i^{\text{CF}} - t_i| > \delta$. We assign a weight $w_i(t_i^{\text{CF}}) \propto \exp(-\widehat{\sigma}^2[y_i(t_i^{\text{CF}})])$ and we apply a weighted loss to minimize the influence of unreliable counterfactuals on the overall loss.

$$\mathcal{L}_{\text{KS}} = \sum_{\substack{i: \\ |t_i^{\text{CF}} - t_i| > \delta}} \frac{e^{-\widehat{\sigma}^2[y_i(t_i^{\text{CF}})]}}{\sum_j e^{-\widehat{\sigma}^2[y_j(t_j^{\text{CF}})]}} \ell(\eta(\Phi(\mathbf{x}_i), t_i^{\text{CF}}), \widehat{y}_i(t_i^{\text{CF}})) \tag{7}$$

We compute the GP quantities $\widehat{y}_i(t_i^{\text{CF}}), \widehat{\sigma}_i^2$ in Eq. 7 inside `stop_gradient`. In practice, the GP is effective only when the outcomes inferred from it outperform those of a baseline model trained solely on the observational dataset. We show that the GP indeed produces better outcomes in Section 5.3. Moreover, we also show that the additional loss on sampled new treatments successfully addresses the discrepancy between the training and counterfactual test distributions, even after suppressing the impact of instances with high variance.

3.3 Estimating Parameters

Fixing GI+GP Parameters The GI and GP layers for inferring the counterfactual outcomes require three parameters δ, σ, and ϵ_{GP}, which we fix based on the validation dataset. The estimation procedure does not involve training and thus is computationally efficient. The δ decides if we estimate $\widehat{y}$ from the GI or the GP, and we fix it to minimize average L2 loss over the validation set. We infer the GP-based mean and variance estimates on the validation dataset at observed treatments for different ϵ_{GP} and σ values and compute the KS loss $\mathcal{L}_{\text{KS}}$ on the validation dataset D_{val} using Eqn 7. In particular, for a sample $(\mathbf{x}, t, y)$ in the validation dataset, its loss is computed on the ground truth label y for the observed treatment t as $\frac{\exp(-\widehat{\sigma}^2[y(t)]) \cdot \ell(y, \widehat{y}(t))}{\sum_{(x',t',y') \in D_{\text{val}}} \exp(-\widehat{\sigma}^2[y'(t')])}$. Finally, we select the parameter values that yield the lowest loss.

Estimation of Φ, η Our final objective function combines the three losses:

$$\min_{\Phi,\eta} (\mathcal{L}_{\text{factual}} + \lambda_{\text{GI}} \mathcal{L}_{\text{GI}} + \lambda_{\text{KS}} \mathcal{L}_{\text{KS}}) \tag{8}$$

where $\lambda_{\text{GI}}, \lambda_{\text{KS}}$ are hyper-parameters that weigh the contributions of the individual loss terms. A brief description of the overall training and inference procedure is described in Algorithm 1.

Algorithm 1: GIKS training

```
Require: Training Data $D = \{(\mathbf{x}_i, t_i, y_i)\}$, Validation Data
    $D_{\text{val}}, P(T), \lambda_{\text{GI}}, \lambda_{\text{KS}}$, # of epochs Epochs, starting epoch
    for GI loss Epoch$_{\text{GI}}$, and for KS loss Epoch$_{\text{GP}}$.
$\Phi, \eta \leftarrow$ Train(VCNET, $D$)
$\delta, \sigma, \epsilon_{\text{GP}} \leftarrow$ FixGIGPParams($D, D_{\text{val}}, \Phi, \eta$)
for $e \in$ [Epochs] do
    loss $\leftarrow$ FactualLoss($D, \Phi, \eta$)
    $t_i^{\text{CF}} \sim P(T)$ for all $i \in [|D|]$
    $S_\delta \leftarrow \{(\mathbf{x}_i, t_i, y_i) : |t_i - t_i^{\text{CF}}| < \delta\}$
    loss $\leftarrow$ loss + GILoss($S_\delta, \{t_i^{\text{CF}}\}, \delta, \Phi, \eta$)
    loss $\leftarrow$ loss + KSLoss($D \backslash S_\delta, \{t_i^{\text{CF}}\}, \epsilon_{\text{GP}}, \Phi, \eta$)
    $\Phi, \eta \leftarrow$ GradDesc(loss)
Return $\Phi, \eta$
```

```
function GILoss($S_\delta, \{t_i^{\text{CF}}\}, \delta, \Phi, \eta$)
for $(\mathbf{x}_i, t_i, y_i) \in S_\delta$ do
    $\widehat{y}_i(t_i^{\text{CF}}) \leftarrow y_i - (t_i - t_i^{\text{CF}}) \frac{\partial \eta(\Phi(\mathbf{x}_i), t)}{\partial t}$
Return $\sum_{(\mathbf{x}_i, t_i, y_i) \in S_\delta} \mathcal{L}(\eta(\Phi(\mathbf{x}_i), t_i^{\text{CF}}), \widehat{y}_i(t_i^{\text{CF}}))$
```

```
function KSLoss($D', \{t_i^{\text{CF}}\}, \epsilon_{\text{GP}}, \Phi, \eta$)
for $(\mathbf{x}_i, t_i, y_i) \in D'$ do
    $D_{\text{NN}}(t_i^{\text{CF}}) \leftarrow$ NNbr($D, \mathbf{x}_i, t_i, \epsilon_{\text{GP}}$) (Eq. 4)
    with stop_gradient:
        $\text{m}_i \leftarrow \mathbb{E}[y_i(t_i^{\text{CF}}) \mid D_{\text{NN}}(t_i^{\text{CF}}), \mathbf{x}_i]$ (Eq. 5)
        $\hat{\sigma}_i^2 \leftarrow \text{Var}[y_i(t_i^{\text{CF}}) \mid D_{\text{NN}}(t_i^{\text{CF}}), \mathbf{x}_i]$ (Eq. 6)
Compute $\mathcal{L}_{\text{KS}}$ using Eq. (7)
Return $\mathcal{L}_{\text{KS}}$
```

4 Related Work

In this section, we briefly review the literature on both discrete and continuous treatment effect estimation.

Discrete Treatment Effect Estimation (DTE) Discrete treatment effect estimation can be categorized into three approaches: (1) *Sample re-weighting* methods (Robins, Rotnitzky, and Zhao 1994; Funk et al. 2011), which adjust counterfactual estimates using inverse propensity scores but can be unstable without calibrated propensity score models. (2) *Feature matching techniques* (Johansson, Shalit, and Sontag 2016; Caliendo and Kopeinig 2008; Rubin 1973; Schwab, Linhardt, and Karlen 2018; Kallus 2020), which infer pseudo targets by aggregating labels of neighboring instances but are sensitive to distance metrics and lack reliability checks. (3) *Regularization-based* methods (Shalit, Johansson, and Sontag 2017; Shi, Blei, and Veitch 2019), such as those using Integral Probability Metrics like Wasserstein distance and Maximum Mean Discrepancy, including Targeted Regularization. Such regularizers are introduced to improve Average Treatment Effect (ATE) estimation. Other approaches include adversarial training methods (Yoon, Jordon, and Van Der Schaar 2018; Ozery-Flato, Thodoroff, and El-Hay 2018), variational autoencoders (Louizos et al. 2017; Rissanen and Marttinen 2021; Lu et al. 2020), and nonparametric Gaussian Process methods that discard the mean estimates and directly minimize the variance of counterfactual predictions (Alaa and Van Der Schaar 2017; Zhang, Bellot, and Schaar 2020).

Continuous Treatment Effect Estimation Existing literature on continuous treatment effect estimation has focused on two aspects: (1) designing better neural architectures, and (2) designing better loss functions and regularizers. The problem of CTE estimation was introduced in (Schwab et al. 2020), that proposed DRNet, that discretizes dosage values and uses separate last layers for each discrete dosage bin while sharing previous layers. VCNet (Nie et al. 2021), on the other hand, avoids discretization by ensuring the smoothness of counterfactual predictions through a trainable spline function. Additionally, TransTEE (Zhang et al. 2022) proposed a Transformer-based representation network specifically designed for text datasets.

VCNet (Nie et al. 2021) introduced a targeted regularizer to address the train-test mismatch and improve ATE estimation accuracy. In (Bellot, Dhir, and Prando 2022), VCNet was extended to use the Hilbert Schmidt Independence criterion as a regularizer for generating treatment-independent embeddings. Another way to enforce independence is by discretizing treatment groups and using an IPM regularizer to make the representations of different treatment groups similar (Wang et al. 2022). TransTEE (Zhang et al. 2022) further extended targeted regularizers to handle continuous treatments with a proposed probabilistic targeted regularizer. While targeted regularizers ensure consistent ATE estimates with asymptotic guarantees, they do not account for Individual Treatment Effect estimation. Another method that performs data augmentation like ours is SciGAN (Bica, Jordon, and van der Schaar 2020), which employs a generative adversarial network (GAN) to generate outcomes for new treatments. However, we will demonstrate the unstable training nature of this GAN-based approach in our experiments.

5 Experiments

Dataset Following the prior work (Nie et al. 2021; Zhang et al. 2022; Bica, Jordon, and van der Schaar 2020), we use five datasets, namely IHDP, NEWS, and TCGA(0-2) on three types of treatments. Across all datasets, $t \in [0, 1]$. Details are deferred to the Appendix.

Methods We compare against six recent state-of-the-art baselines, *i.e.*, TARNet (Shalit, Johansson, and Sontag 2017), DRNet (Schwab et al. 2020). SciGAN (Bica, Jordon, and van der Schaar 2020), TransTEE (Zhang et al. 2022) and VCNet+TR (Nie et al. 2021). Details of methods in Appendix.

Hyper-Parameter Estimation We allocate 30% samples as validation dataset to tune hyperparameters. Note, we depend only on *factual error* and do not require counterfactual supervision even in the validation dataset. GIKS has three hyperparameters: learning rate, λ_{GI}, λ_{GP} that are optimized via grid search on factual error of the validation dataset. Further, the GP employs a cosine kernel. We use a batch size of 128, the AdamW optimizer, and early stopping based on factual error on the validation dataset. The results of hyperparameter tuning are presented in Table 3.

Evaluation Metric (CF Error) Following existing literature, we evaluate performance using CF Error, short

	TARNet	DRNet	SciGAN	TransTEE	VCNet+TR	VCNet+HSIC	GIKS
TCGA-0	1.673 (0.00)	1.678 (0.00)	2.744 (0.00)	0.164 (0.25)	0.163 (0.31)	0.164 (0.29)	**0.152**
TCGA-1	1.417 (0.00)	1.465 (0.00)	0.907 (0.00)	0.146 (0.00)	0.098 (0.03)	0.096 (0.08)	**0.080**
TCGA-2	3.365 (0.00)	3.396 (0.00)	1.359 (0.01)	0.201 (0.00)	0.152 (0.00)	0.144 (0.02)	**0.127**
IHDP	2.731 (0.00)	3.068 (0.00)	–	2.266 (0.00)	2.263 (0.00)	1.961 (0.09)	**1.891**
NEWS	1.126 (0.00)	1.163 (0.00)	–	1.239 (0.00)	1.107 (0.00)	1.104 (0.00)	**1.079**

Table 1: Comparison of GIKS with baselines. on CF error. The table includes the mean performance and within brackets, the p-values obtained from one-sided paired t-tests with GIKS as the base. Values less than 0.05 indicate statistically significant gains. ”–” denotes models that did not converge. GIKS outperforms all the baselines across all datasets, and the results are statistically significant for the majority of them.

	TARNet		DRNet		TransTEE	
	Baseline	GIKS	Baseline	GIKS	Baseline	GIKS
TCGA-0	1.678 ± 0.027	**1.077 ± 0.034**	1.673 ± 0.036	**1.073 ± 0.028**	**0.164 ± 0.027**	0.165 ± 0.032
TCGA-1	1.465 ± 0.039	**0.555 ± 0.011**	1.417 ± 0.049	**0.556 ± 0.013**	0.146 ± 0.024	**0.132 ± 0.021**
TCGA-2	3.396 ± 0.059	**0.746 ± 0.045**	3.365 ± 0.079	**0.747 ± 0.041**	0.201 ± 0.011	**0.172 ± 0.179**
IHDP	3.068 ± 0.126	**3.037 ± 0.227**	2.731 ± 0.333	**2.532 ± 0.169**	2.266 ± 0.182	**2.023 ± 0.193**
NEWS	1.163 ± 0.055	**1.159 ± 0.088**	**1.126 ± 0.069**	1.129 ± 0.135	1.239 ± 0.120	**1.176 ± 0.179**

Table 2: Performance comparison of GIKS with base model architecture adopted from three other state of the art methods, *viz*, TARNet, DRNet and TransTEE on the CF error.

	TCGA(0-2)	IHDP	NEWS
lrn rate	10^{-4}	10^{-2}	10^{-3}
λ_{GI}	10^{-1}	10^{-4}	10^{-2}
λ_{KS}	10^{-2}	10^{-1}	10^{-4}

Table 3: The hyperparameters.

for counterfactual estimation error, that measures the prediction accuracy for arbitrary treatments applied on test instances, thus making it suitable for the ICTE problem. Given N test instances, we define the CF error as: $\sqrt{\frac{1}{N}\sum_{i=1}^{N}\int_{t_i^{\text{CF}}=0}^{1}(y_i(t_i^{\text{CF}})-\widehat{y}_i(t_i^{\text{CF}}))^2 P(t_i^{\text{CF}})dt_i^{\text{CF}}}$. The error integrates over treatments sampled from $P(T)$ applied to the test instances. In practice, since it is difficult to determine the test time treatment distribution $P(T)$, our default is using a uniform distribution, as followed in earlier work (Bica, Jordon, and van der Schaar 2020), and we present ablations on other candidates of $P(T)$.

5.1 Comparison with SOTA Methods

Table 1 compares our method against all state-of-the-art CTE methods (Section 5) based on CF Error. We make the following observations: **(1)** GIKS consistently outperforms all baselines on CF error across all datasets, highlighting the suitability of our loss function for ICTE estimation. The statistically significant gains in performance, as indicated by the computed p-values from a one-sided paired t-test with GIKS as the base, further support the superiority of our approach, except for the TCGA-0 dataset where performance is comparable to the next competing baselines. **(2)** SciGAN, despite incorporating instance-level counterfactual losses like our approach, demonstrates poor performance due to the challenges associated with training the min-max objectives in adversarial training. Our experiments revealed instances where the error significantly increased for specific dataset seeds, resulting in high result variance. The lack of a control mechanism akin to our GP variance to filter unreliable counterfactual supervision prevented model convergence when the counterfactual supervision provided by the generator was flawed. **(3)** DRNet and TARNet suffer from poor performance due to their discretization of treatments, which leads to their η network being less sensitive to changes in t. **(4)** VCNet with two regularizers: Targeted Regularizer and HSIC are both worse than GIKS. Although HSIC is a better regularizer for the ICTE problem, it still operates at a distribution level rather than at an instance level like GIKS.

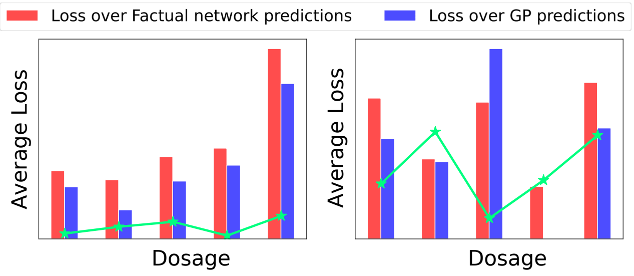
5.2 GIKS with Different Base Architectures

The model-agnostic nature of GIKS allows it to be integrated with various base architectures. While VCNet is chosen for its effectiveness with continuous treatments, we explore the potential of GIKS with other networks such as TARNet, DRNet, and TransTEE next. Our experiments, presented in Table 2, show that GIKS enhances the performance of all the base networks.

5.3 Why GIKS Works?

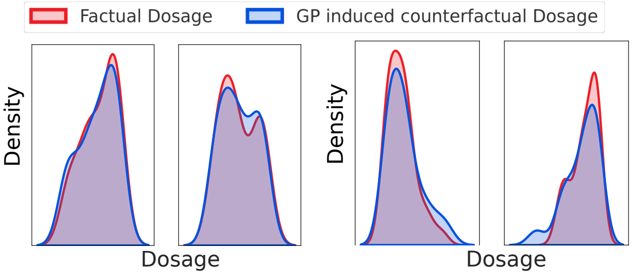
To see why our approach produces better counterfactual estimates, we conduct experiments using the IHDP dataset, and a synthetic dataset taken from Nie et al. (2021), to answer the following questions:

- Are inferred counterfactuals using neighbors in Φ and T space more accurate than those obtained from a baseline model that is trained solely on observational dataset D?
- For a given individual $\mathbf{x}$, do our augmented loss help reduce the divergence $D(P(\mathbf{x},t)||P(\mathbf{x})P(t))$ between training and test distribution?



(a) IHDP (b) Synthetic

Figure 1: Losses on Counterfactuals



(a) IHDP (b) Synthetic

Figure 2: Training Dosage distribution

We answer the first question by contrasting the performances of two models: (1) a baseline factual model trained solely on the observational dataset D using factual loss $\mathcal{L}_{\text{factual}}$, and (2) our GIKS model trained using algorithm 1. Now, for GIKS to work, we need GP to produce counterfactual estimates that are more accurate than the factual model. To assess this, we compare the losses of the factual model with the losses of GP estimates obtained using Eq. 5 for training instances at randomly sampled new treatments. The results in Figure 1 demonstrate that GP produces more accurate counterfactual estimates than the baseline's η network, except for the middle bin $(0.4, 0.6]$ in the synthetic dataset, which has limited instances. The line plot in Figure 1 shows the distribution of examples in each bin. By performing a one-sided paired t-test, we confirm that GP's losses are statistically significantly lower than the factual losses with a p-value of 0 for both datasets, providing further confidence in GP's higher accuracy.

To answer the second question, we compare the treatment distributions used to train the factual model and GIKS for different individuals in the dataset. Figure 2 presents the results for both datasets, showcasing the treatments at which losses were imposed during training for two arbitrarily chosen candidates and their 30 nearest neighbors. We observe that GIKS reduces the skew in the treatment distribution, leading to a lower divergence $D(P(\mathbf{x}, t)||P(\mathbf{x})P(t))$. To further validate this observation, we compute the HSIC metric, which resulted in divergence values of 8.30 (0.94) for the factual model of synthetic (IHDP) and 4.9 (0.63) for GIKS of Synthetic (IHDP) dataset. These results, along with the previous findings, provide insights into the effectiveness of GIKS for counterfactual inference.

5.4 Ablation Study

Impact of the Three Losses on GIKS Performance In this experiment, we analyze the impact of each of the three proposed losses in GIKS by measuring the CF error achieved for models that are trained on different combinations of GIKS losses until convergence. The results are summarized in Table 4. First, observe that neither GI, nor GP is superior over the other, and this lets us conclude that both our loss components have a non-trivial effect on the performance of GIKS. Second, the GI loss alone manages to provide gains over VCNet and complements the smoothness provided by the spline parameterization of the VCNet model. While VCNet focuses on smooth parameter variations of the η network with t, our $\mathcal{L}_{\text{GI}}$ ensures that the predictions of the η network also vary smoothly with t. These two methods work together to achieve a smooth ADRF at an instance level. We observe that $\mathcal{L}_{\text{GI}}$ helps in smoothing the predicted Dose-Response Function for treatments close to the observed treatments, as elucidated in Figure 3 for two training instances.

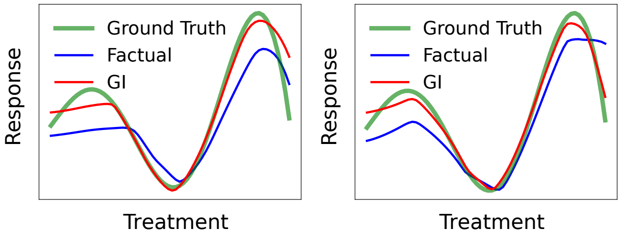


Figure 3: IHDP Individualised Dose-Response Function

Dataset	$\mathcal{L}_{\text{fct}}$	$+\lambda_{\text{GI}}\mathcal{L}_{\text{GI}}$	$\lambda_{\text{KS}}\mathcal{L}_{\text{KS}}$	GIKS
TCGA-0	0.18 (0.10)	0.17 (0.26)	0.17 (0.21)	**0.15**
TCGA-1	0.09 (0.08)	0.09 (0.62)	0.09 (0.63)	**0.09**
TCGA-2	0.17 (0.03)	0.17 (0.06)	0.16 (0.01)	**0.12**
IHDP	2.05 (0.00)	1.91 (0.32)	1.96 (0.10)	**1.89**
NEWS	1.09 (0.00)	1.08 (0.41)	1.08 (0.23)	**1.07**

Table 4: CF error for the models trained using different combinations of our loss components.

	CF error
NTK	1.925 (0.35)
Dot product	1.882 (0.43)
cosine	**1.842**

Table 5: GIKS performance using different kernels over 5 seeds of IHDP dataset.

	IHDP	NEWS
Marginal	1.902 (0.42)	1.086 (0.09)
IP	1.903 (0.41)	1.080 (0.36)
Uniform	**1.891**	**1.079**

Table 6: Sampling strategies for $P(T)$.

Kernel Exploration We study the impact of different types of kernels on the performance of GIKS. We experimented with three kernels: cosine kernel $K(\Phi(\mathbf{x}), \Phi(\mathbf{x}')) = \frac{\Phi(\mathbf{x})^\top \Phi(\mathbf{x}')}{||\Phi(\mathbf{x})||||\Phi(\mathbf{x})||}$, dot product kernel $K(\Phi(\mathbf{x}), \Phi(\mathbf{x}')) = \Phi(\mathbf{x})^\top \Phi(\mathbf{x}')$, and finally NTK kernel $K(\Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j)) = \nabla_{\eta,\Phi}\eta(\Phi(\mathbf{x}_i), t_i)^\top \nabla_{\eta,\Phi}\eta(\Phi(\mathbf{x}_j), t_j)$. Table 5 summarizes the results for the IHDP dataset averaged over 5 seeds. We present the p-values of one-sided paired t-test with cosine kernel as the base within bracket. We observe that: (1) Cosine kernel performs better than Dot-product kernel, perhaps because Cosine is invariant to the scale of the embeddings. (2) Even though the NTK kernel defines a joint kernel over $\mathcal{X}$ and $\mathcal{T}$ space, unlike GIKS that used a two-stage kernel, computing the NTK Kernel is both computation and memory intensive.

Impact of Alternative $P(T)$ We experimented with alternative distributions for sampling new treatments during training: (1)*Marginal treatment distribution*, where we sample treatments from the empirical marginal treatment distribution in the observational dataset. (2)*Inverse Propensity (IP) distribution*, where we discretized the treatments into 10 bins and trained a propensity model $\pi : \mathcal{X} \rightarrow [10]$ that predicts the treatment bin. Then we sample treatments from a bin chosen with a probability $P(\bullet \,|\, \mathbf{x}) = \frac{1}{\pi(\bullet \,|\, \mathbf{x})}$. To make the results comparable, Table 6 presents the CF error tested on uniform distribution across these three training strategies. We also report p-values of the one-sided paired t-test with default sampling. We observe that GIKS performance is similar across all the sampling strategies, that is possibly because (1) For the IHDP and news dataset, their marginal treatment distribution is close to uniform (3) For IP sampling, the propensity model achieved an accuracy of 100% in predicting the correct bin for both datasets. The inverted propensity scores gave negligible weight to the predicted bin and approximately equal weight to the other bins leading to close to uniform sampling.

	TransTEE	VC+TR	VC+HSIC	GIKS
CF Error	**0.42 ± 0.01**	0.43 ± 0.01	0.43 ± 0.01	**0.42 ± 0.01**
Policy Err.	0.22 ± 0.03	0.19 ± 0.02	0.18 ± 0.00	**0.17 ± 0.01**
Rec. Acc.	0.59 ± 0.02	0.60 ± 0.02	0.62 ± 0.04	**0.63 ± 0.01**

Table 7: Performance of GIKS vs. baselines on recourse for image setting for skin lesion diagnosis. We report mean ± std. deviation averaged over 5 seeds.

5.5 Case Study

We conducted two case studies illustrating the application of treatment effect estimation to Algorithmic Recourse where the goal is to predict the treatment that yields the best outcome. We present the skin lesion diagnosis experiment in this section and defer the insulin prediction experiment to regulate glucose levels to the appendix.

Algorithmic Recourse for Mobile Skin Lesion Diagnosis We consider an application where users submit skin images to a diagnostic classifier designed for lesion detection. In the event of a user uploading a low-quality image, the classifier might yield an inaccurate diagnosis. Our objective is to guide users in adjusting their image capture settings to enhance diagnostic accuracy. We employed a commonly used skin lesion detection dataset, featuring seven labels, with the classifier predicting a distribution across these seven categories. Our goal in algorithmic recourse is to recommend image settings for which the classifier confidence — probability of the predicted class — is high. As part of the setting (treatment), we consider brightness level of the image. For training the model for counterfactual inference we created an observation dataset D, where each instance is a 3-tuple $(t_i, \phi(\mathbf{x}_i), \rho_i)$. Here, $\mathbf{x}_i$ is an initial image, $\phi(\mathbf{x}_i)$ is its representation, t_i denotes the treatment applied to produce post-treatment image $\mathbf{x}'_i$, and ρ_i reflects the classifier's confidence on the treated image. We introduced selection bias by making the observed treatments depend on $\phi(\mathbf{x}_i)$. Testing the classifier on a test set with treatments sampled uniformly from $[-0.5, 0.5]$, we observed an accuracy of 44%. We evaluate the performance on VCNet architecture of baselines vs. GIKS using Recourse Accuracy which is defined as the accuracy achieved on the test dataset after treating the test images with predicted optimal dosages. We report the recourse performance comparing GIKS with other competing baselines in Table 7. We observed that GIKS achieves the highest recourse accuracy of $63 \pm 0.02\%$. We also report dosage policy error which is the difference between the accuracy achievable at the best brightness setting, and the accuracy at the recommended setting by the trained model. Even on this metric, our model performs better.

6 Conclusion

We addressed estimating Individualized Continuous Treatment Effects from observational data, where treatments are confounded with covariates. Our method aims to reduce the mismatch between observed data distribution and the independence needed for counterfactual estimation by sampling new treatments for training instances. We devised two strategies for synthesizing pseudo targets and applying instance-specific counterfactual losses. Experiments on benchmark ICTE datasets showed statistically significant gains over other baselines. We also presented experimental results on two potential medical applications. Future work could include a more thorough investigation of these applications and extending our approach to cases where the overlap assumption is violated as recently highlighted in (Jesson et al. 2022).

References

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

Alvarez Melis, D.; and Jaakkola, T. 2018. Towards Robust Interpretability with Self-Explaining Neural Networks. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

Arjovsky, M.; Bottou, L.; Gulrajani, I.; and Lopez-Paz, D. 2019. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Bellot, A.; Dhir, A.; and Prando, G. 2022. Generalization bounds and algorithms for estimating conditional average treatment effect of dosage.

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

Caliendo, M.; and Kopeinig, S. 2008. Some practical guidance for the implementation of propensity score matching. *Journal of economic surveys*, 22(1): 31–72.

Funk, M. J.; Westreich, D.; Wiesen, C.; Stürmer, T.; Brookhart, M. A.; and Davidian, M. 2011. Doubly robust estimation of causal effects. *American journal of epidemiology*, 173(7): 761–767.

Jacot, A.; Gabriel, F.; and Hongler, C. 2018. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31.

Jesson, A.; Douglas, A.; Manshausen, P.; Meinshausen, N.; Stier, P.; Gal, Y.; and Shalit, U. 2022. Scalable sensitivity and uncertainty analysis for causal-effect estimates of continuous-valued interventions. *arXiv preprint arXiv:2204.10022*.

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

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

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

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

Nasery, A.; Thakur, S.; Piratla, V.; De, A.; and Sarawagi, S. 2021. Training for the Future: A Simple Gradient Interpolation Loss to Generalize Along Time. *Advances in Neural Information Processing Systems*, 34: 19198–19209.

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

Ozery-Flato, M.; Thodoroff, P.; and El-Hay, T. 2018. Adversarial Balancing for Causal Inference. *ArXiv*, abs/1810.07406.

Pearl, J.; and Press, C. U. 2000. *Causality: Models, Reasoning, and Inference*. Cambridge University Press. ISBN 9780521773621.

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

Robins, J. M.; Rotnitzky, A.; and Zhao, L. P. 1994. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427): 846–866.

Rubin, D. B. 1973. Matching to remove bias in observational studies. *Biometrics*, 159–183.

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

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

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

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

Titsias, M. 2009. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial intelligence and statistics*, 567–574. PMLR.

Wang, X.; Lyu, S.; Wu, X.; Wu, T.; and Chen, H. 2022. Generalization bounds for estimating causal effects of continuous treatments. *Advances in Neural Information Processing Systems*, 35: 8605–8617.

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

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

Zhang, Y.-F.; Zhang, H.; Lipton, Z. C.; Li, L. E.; and Xing, E. P. 2022. Exploring Transformer Backbones for Heterogeneous Treatment Effect Estimation.