Replidation GANITE: Estimation of Indivisualized Treatment Effects Using Generative Adversarial Nets

Yohei Nishimura

University of Wisconsin-Madison

Abstract

The report focuses on replicating and extending the GANITE model for estimating individualized treatment effects using generative adversarial nets. It addresses the challenge of causal effect estimation, particularly Individualized Treatment Effect (ITE) estimation, using a Generative Adversarial Network (GAN) framework. The report includes a detailed explanation of the problem formulation, literature review, dataset description, performance metrics, network architecture, and training algorithm of the GANITE model. The experiments and results section compares GANITE with other methods like OLS, k-NN, and BART, emphasizing its superior performance in estimating heterogeneous effects and average treatment effects. The extension section explores improvements through deeper architecture, advanced optimization algorithms, and simultaneous training. The report concludes with a discussion on the potential of advanced computational techniques in predictive accuracy and suggests directions for future research as wee as the potential limitation of this method.

Replidation GANITE: Estimation of Indivisualized Treatment Effects Using Generative Adversarial Nets

Context and motivation

Causal effect estimation is a challenging problem since it includes the counter factual outcome. Therefore, researchers have focused on the expectation of the treatment effect such as Average Treatment Effect (ATE) and Average Treatment Effect on the Treated (ATT). However, the direct estimation of Individualized Treatment Effect (ITE) is more preferable, if the challenging task is overcome where we are supposed to calculate an unbiased estimator of an individual's potential outcomes without access to the counter factual data.

The target paper, named GANITE: Estimation of individualized treatment effects using generative adversarial nets (Yoon, Jordon, and Van Der Schaar (2018)), attempts to resolve this contradictory conundrum and does so at a certain level. The motivation of the paper is that the replication of the novel approach to estimate the causal effect using observational data which has the natural shortcoming, that is, treatment selection bias to be used for estimation, causing the trained model from the biased data not to be generalized well to the entire population.

In order to address the challenge, the research utilizes Generative Adversarial Network (GAN, Goodfellow et al. (2020)) generating the counterfactuals to be used the calculation of ITE; based on the feature data, treatment variables, and factual outcome, the generator network called $\boldsymbol{G}$ artificially fabricates the counter factual outcome.

Literature Reviews

In the field of Individualized Treatment Effect (ITE) estimation, prior research has employed various methodologies. One common approach is to learn a separate model for each treatment, but this method often overlooks selection bias, resulting in models biased towards their specific treatment populations. Another widespread strategy involves the treatment as a feature within a single model. This approach adjusts for the mismatch between the sample distribution and the distributions of

treated and control groups to mitigate selection bias. Techniques used here include tree-based models, as seen in works by Chipman, George, and McCulloch (2012), Wager and Athey (2018), and others; doubly-robust methods; k-nearest neighbor; propensity and matching-based methods (Crump, Hotz, Imbens, and Mitnik (2008)); and deep learning approaches Johansson, Shalit, and Sontag (2016); Shalit, Johansson, and Sontag (2017). A significant challenge in this method is balancing the need for predictive information while minimizing biased information, especially pertinent in medical settings where treatment decisions are often based on predictive features.

Additionally, some researchers have explored multi-task model approaches. This involves using multi-task neural networks to simultaneously estimate selection bias, controlled outcomes, and treated outcomes, integrating shared layers across these tasks. Notably, Alaa, Weisz, and Van Der Schaar (2017) implemented this using multi-task neural nets, while Alaa and Van Der Schaar (2017) applied a Gaussian Process approach. These approaches are distinctive for their ability to incorporate counterfactuals and provide confident estimates through credible intervals, achieved by incorporating posterior distributions in the model learning process.

Problem Formulation

The research is assumed to be in accordance with the Rubin-Neyman causal model Rubin (2005). Consider a feature space X of dimension s and a set of possible outcomes Y. The joint distribution μ on $X \times \{0,1\}^k \times Y^k$, where k is the number of possible treatments, is given. Let $(X, T, Y) \sim \mu$, with $X \in X$ as the s-dimensional feature vector, $T \equiv (T_1, \ldots, T_k) \in \{0,1\}^k$ as the treatment vector, and $Y \equiv (Y_1, \ldots, Y_T) \in Y^k$ as the vector of potential outcomes or the ITE. It is assumed that there is exactly one non-zero component in T, denoted by η. The marginal distribution of X is μ_X, and $\mu_Y(x)$ is the conditional distribution of Y given $X = x$, marginalized over T.

Two assumptions are introduced about distribution μ in the Rubin-Neyman causal model as a strong ignorability:

1. **Overlap**: For all $x \in X$ and all $i \in \{1, \ldots, k\}$, it holds that

$0 < P(T_i = 1|X = x) < 1$. This ensures a non-zero probability of receiving any treatment i at every point in the feature space.

2. **Unconfoundedness**: Conditional on X, the potential outcomes Y are independent of T, denoted as $Y \perp T|X$. This implies no unmeasured confounding, allowing $\mu_Y(x)$ to be independent of T.

In an empirical study, samples of (X, T, Y_η) are observed, forming the dataset $D = \{(x^{(i)}, t^{(i)}, y_\eta^{(i)})\}_{i=1}^N$. The observed component of the potential outcome vector, corresponding to the assigned treatment, is called the *factual* outcome, and the unobserved potential outcomes are referred to as *counterfactuals*. We denote the factual outcome as y_f^i, and vector of counterfactuals as y_{cf}^i.

The goal is to draw samples from $\mu_Y(x)$ for any $x \in X$. The performance of the generator $I(x)$ is measured using two metrics, depending on whether $k = 2$ (binary treatments) or $k > 2$ (multiple treatments).

In our replication, we use $k = 2$ semi-simulated data called **Twins** to be explained in the following chapter. Thus, we use the expected Precision in Estimation of Heterogeneous Effects, ϵ_{PEHE} introduced in Hill (2011), given by:

$$\epsilon_{PEHE} = \mathbb{E}_{x\sim\mu_x}[(\mathbb{E}_{y\sim\mu_{Y(x)}}(y_1 - y_0) - \mathbb{E}_{\hat{y}\sim I(x)}(\hat{y}_1 - \hat{y}_0))^2]$$

Replication

We will provide the replication detail including dataset, network architectures, training algorithm, performance metrics, and the empirical results comparing to the results by the original paper.

Dataset

The performance evaluation of the GANITE algorithm utilizes a combination of semi-synthetic and real-world datasets. This approach is necessitated by the inherent difficulty in evaluating causal inference algorithms on real-world datasets, where ground

truth for counterfactual outcomes is typically unobservable. We select Twins, one of the datasets employed in the original paper, which was invented in the other research paper (Almond, Chay, and Lee (2005)). For each twin-pair we obtained 30 features relating to the parents, the pregnancy and the birth: marital status; race; residence; number of previous births; pregnancy risk factors; quality of care during pregnancy; and number of gestation weeks prior to birth.

A detailed description of the Twins dataset is provided in the original paper; thus we summarise it here. Derived from US birth records between 1989-1991, the dataset focuses on twin births, defining the treatment $t = 1$ for the heavier twin and $t = 0$ for the lighter twin. The primary outcome is the one-year mortality. The dataset includes 30 features related to the parents, pregnancy, and birth. Only twins weighing less than 2kg and without missing features are included, resulting in 11400 twin pairs. The mortality rate for the lighter twin is 17.7% and for the heavier twin 16.1%. This dataset allows for the observation of both treatment cases (heavier and lighter twin) in each pair, providing a unique ground truth for individualized treatment effects. To simulate an observational study, a selection bias is introduced: $t|x \sim Bern(Sigmoid(w^T x + n))$ where $w^T \sim U((-0.1, 0.1)^{30\times 1})$ and $n \sim N(0, 0.1)$.

Performance Metrics

In the original paper, for Twin dataset, they use estimated empirical Precision in Estimation of Heterogeneous Effect (PEHE). In cases where both factual and counterfactual outcomes are observed but their underlying distribution is unknown, as in the Twins dataset, they use:

$$\hat{\epsilon}_{PEHE} = \frac{1}{N}\sum_{i=1}^{N}\left[\left(y_1^{(i)} - y_0^{(i)}\right) - \left(\hat{y}_1^{(i)} - \hat{y}_0^{(i)}\right)\right]^2.$$

Additionally, they introduced average treatment effect (ATE) as another metrics for the performance measurement. Again, since they use the Twin data where both factual and counterfactual outcomes are observed but the underlying distribution is

unknown, the difference ATE is defined as:

$$\hat{\epsilon}_{ATE} = \left(\frac{1}{N} \sum_{i=1}^{N} y^{(i)} - \frac{1}{N} \sum_{i=1}^{N} \hat{y}^{(i)} \right)^2$$

In our replication, we will use these two metrics to evaluate our model.

Networks

The primary goal of GANITE is to generate potential outcomes for a given feature vector x. The lack of direct access to counterfactual outcomes necessitates an approach to estimate these outcomes indirectly. GANITE utilizes a counterfactual generator G and an ITE (Individualized Treatment Effect) generator I within a conditional GAN framework to achieve this. Overall, the architecture is composed of two blocks: counterfactual block and ITE block. Figure 1 from the original paper shows the concept of the two networks.

Counterfactual Block. The GANITE framework utilizes generative adversarial networks to estimate potential outcomes in scenarios where direct observation of counterfactuals is not possible. It comprises two main blocks: the counterfactual generator and the ITE generator, each optimized through a combination of adversarial and supervised learning techniques.

The Counterfactual Generator G in the GANITE framework plays a pivotal role in generating potential outcome vectors $\tilde{y}$ from a given feature vector x, treatment vector t, and observed factual outcome y_f. It is defined as $G(x, t, y_f) = g(x, t, y_f, z_G)$, where z_G is a noise vector sampled from a uniform distribution $U((-1, 1)^{k-1})$. The function g is designed to map the input space into the potential outcome space, simulating the distribution of counterfactual outcomes.

The Counterfactual Discriminator D_G complements the generator by assessing the realism of the generated outcomes. It receives inputs $(x, \bar{y})$ and outputs a probability vector indicating the likelihood $[0, 1]$ that each component of $\tilde{y}$ is factual. The discriminator's objective is to identify factual components among the generated outcomes, thus guiding G to improve the generation of realistic counterfactuals.

ITE Block. The ITE Generator I, on the other hand, focuses on generating potential outcome vectors $\hat{y}$ using only the feature vector x. It is mathematically represented as $I(x) = h(x, z_I)$, where z_I is a noise vector sampled from $U((-1, 1)^k)$. The goal of I is to approximate the distribution of potential outcomes as closely as possible, independent of the treatment vector.

The ITE Discriminator D_I functions in tandem with I, evaluating the generated potential outcomes against the complete dataset $\tilde{D}$. It takes a pair (x, y^*) as input and returns a scalar value representing the probability that y^* was drawn from the dataset $\tilde{D}$ rather than being generated by I. This adversarial setup enables I to refine its generation process, aiming to produce outcomes indistinguishable from real data.

Optimization problem and Training Algorithm

We will explain the empirical loss functions employed for optimizing the components of the GANITE framework. The optimization process is vital for ensuring the effective performance of the model in individualized treatment effect estimation.

The counterfactual generator G and its discriminator D_G are involved in a minimax game, formulated as:

$$\min_G \max_{D_G} \mathbb{E}_{(x,t,y_f)\sim\mu_f} \mathbb{E}_{z_G\sim U((-1,1)^k)} [t^T \log D_G(x, \tilde{y}) + (1-t)^T \log(1 - D_G(x, \tilde{y}))]$$

where log is performed element-wise and T denote the transpose operator.

The empirical objective of the minimax problem for the Counterfactual Generator G and its Discriminator D_G is defined based on the equation above. The objective function V_{CF} for a sample $x^{(i)}, t^{(i)}, \bar{y}^{(i)}$ is given by:

$$V_{CF}(x^{(i)}, t^{(i)}, \bar{y}^{(i)}) = {t^{(i)}}^T \log(D_G(x^{(i)}, \bar{y}^{(i)}) + (1 - {t^{(i)}}^T \log(1 - D_G(x^{(i)}, \bar{y}^{(i)}))$$

Additionally, a supervised loss L_{GS} is introduced to ensure the generated factual outcome closely matches the observed factual outcome:

$$L_S^G(y_f^{(i)}, \tilde{y}_\eta^{(i)}) = (y_f^{(i)} - \tilde{y}_\eta^{(i)})^2.$$

With the two objective functions, the optimization of G and D_G is performed iteratively with k_G minibatches:

$$\begin{aligned} \min_{D_G} \quad & -V_{CF}(x^{(i)}, t^{(i)}, \tilde{y^i}) \\ \min_{G} \quad & V_{CF}(x^{(i)}, t^{(i)}, \tilde{y^i}) - \alpha L_S^G(y_f^{(i)}, \tilde{y}_\eta^{(i)}) \end{aligned}$$

where $\alpha \geq 0$ is a hyper-parameter.

On the other hand, the ITE generator I and its discriminator D_I follow a minimax criterion similar to the counterfactual block, but with access to the complete dataset $\tilde{D}$:

$$\min_I \max_{DI} \mathbb{E}_{x \sim \mu_X} \mathbb{E}_{y^* \sim \mu_Y(x)} [\log DI(x, y^*)] + \mathbb{E}_{y^* \sim I(x)} [\log(1 - DI(x, y^*))]$$

After training the Counterfactual Block, the ITE Block consisting of the ITE Generator I and its Discriminator D_I is optimized. The empirical objective of the minimax problem for I and D_I is based on the equation above by using a binary cross entropy loss:

$$V_{ITE}(x^{(i)}, \bar{y}^{(i)}, \hat{y}^{(i)}) = \log(DI(x^{(i)}, \bar{y}^{(i)})) + \log(1 - DI(x^{(i)}, \hat{y}^{(i)}).$$

For the optimization, supervised losses are introduced for binary treatments as:

$$L_S^I(\bar{y}^{(i)}, \hat{y}^{(i)}) = ((\bar{y}_1^{(i)} - \bar{y}_0^{(i)}) - (\hat{y}_1^{(i)} - \hat{y}_0^{(i)}).$$

With the two objective functions, the optimization of I and D_I is performed with k_I minibatches:

$$\begin{aligned} \min_{D_I} \quad & -V_{ITE}(x^{(i)}, \bar{y}^{(i)}, \hat{y}^{(i)}) \\ \min_{I} \quad & V_{ITE}(x^{(i)}, \bar{y}^{(i)}, \hat{y}^{(i)}) + \beta L_S^I(\bar{y}^{(i)}, \hat{y}^{(i)}) \end{aligned}$$

where $\beta \geq 0$ is a hyper-parameter.

Experiments and Results

We implemented GANITE algorithm using PyTorch. The code is on GitHub `https://github.com/YorkNishi999/ganite_pytorch`. Basically, our implimentation

is along with the algorithm provided in the original paper, except for the ITE block. In our implementation, because the loss of the discriminator DI is included in the V_{ITE}, we only calculate V_{ITE} to optimize the Inference net singly. A set of hyperparameter for the model is defined along with the Appendix of the original paper; initialization is Xavier Initialization for weight and Zero Initialization for bias, optimization is Adam (Kingma and Ba (2014)), batch size is 128, depth of layers is 5, hidden state dimention is 8, and α and β are both 2.

To compare the replicated results from GANITE, some of the estimations by other methods described in the paper are also reproduced; least square regression using treatment as a feature (OLS), k-nearest neighbor (k-NN), and Bayesian additive regression trees (BART). We used the existing Python packages for the other estimations; we utilized scikit-learn packages, while bartpy is used for the estimation of BART. Additionally, regarding OLS and k-NN, we employed Monte Carlo method; we randomly divided the dataset into training and test data as the ratio of 8:2 and iterated 1000 times, averaging all the results.

The table 1 shows the results. Based on the hyperparameter described in the paper, we have almost achieved to replicate the paper. GANITE is the best model among four to extimate in-sample metrics while it has acieved the second best for out-sample result. Our GANITE model are trained by 5000 epochs with the same set of hyperparameters shown in the original paper.

Extension

This section focuses on the ablation study for the extension of the original paper, mainly to figure out the new, improved architecture to enhance the results. The influential elements are 1) the architecture of the neural net and 2) optimization algorithm, showing how it improves the results. Moreover, we will change the training algorithm as all the network are optimized simultaneously.

Specifically, we propose the new architecture and training algorithm including 1) more deeper architecture with dropout layers, 2) usage of the advanced optimization

algorithm, and 3) simultaneous training between generator and inference net.

Architecture

The architecture of the neural net highly impacts on the results in deep learning methodologies. In the original paper, simple multi-layer perceptron (MLP) are employed. Therefore, we will try deeper MLP beyond the original paper such as 15 with dropout layers that effectively enhance the results in deep architecture. Additionally, since a deeper MLP network performs well with dropout layers, we will introduce the layers between a MLP layer and an activation function (Srivastava, Hinton, Krizhevsky, Sutskever, and Salakhutdinov (2014)).

Optimization Algorithm

Loshchilov and Hutter (2017) proposed the advanced optimization algorithm named AdamW, which is improved from Adam that is used in the original paper.

AdamW modifies from Adam the way weight decay is integrated into the optimization process. Instead of incorporating weight decay into the parameter updates as in Adam, AdamW decouples the weight decay from the optimization steps. This allows for a more consistent and effective application of weight decay, as it is applied directly to the weights and independent of the adaptive learning rate adjustments, leading to improved training stability and performance.

Training Algorithm

Larsen, Sønderby, Larochelle, and Winther (2016) introduced the method combining GAN with VAE (Variational Auto Encoder), mixing training algorithm GAN and VAE simultaneously. Inspired by the algorithm, we will train Generator/Discriminator in the counterfacctual block and Inference Net in the ITE block in the same epoch, not sequentially.

Results

Table 2 shows the ablation study and final results. It shows the results from five models: 1) original, 2) the model with AdamW and 5-depth layer, 3) the model with drop out layers, Adam, and 5-depth layer, 4) the model with AdamW, 15-depth layer, and simultaneous algorithm, and 5) the model with AdamW, dropout layers, 15-depth layer, and simultaneous algorithm. In terms of the most important metric, PEHE with 'Out-sample' (test data), the fifth model achieved the best score, which is equal to the original paper result as the point estimation, and better than it as the standard error.

Discussion

The results from our reproductive experiments by Table 1, highlight the enhanced performance of the GANITE model over traditional methods like OLS, k-NN, and BART. Notably, the GANITE model demonstrated superior precision in the estimation of heterogeneous effects ($\hat{\epsilon}_{PEHE}$) using in-sample data, which is compatible with the original paper. The extensions introduced, as detailed in comparison with Table 2, including deeper architecture, advanced optimization algorithms, and simultaneous training, improved $\hat{\epsilon}_{PEHE}$ using out-sample data. These improvements underscore the potential of advanced computational techniques in enhancing predictive accuracy.

The usage of real-world data in applications is one of the challenges to be overcome for the broader employment of GANITE. Twin data is critical to generate counterfactuals through simulation. The training of GANITE heavily relies on supervised learning based on this generated data. Therefore, to use the algorithm for the actual verification of causal effects, it is necessary to construct a simulation algorithm that can produce convincing counterfactuals. Consequently, it may take time to replace parametric models that estimate models directly from data.

Also, we have not achieved a significant improvement, suggesting the need for further research to optimize model performance by different architectures or training algorithms to improve the GANITE model.

References

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

Alaa, A. M., Weisz, M., & Van Der Schaar, M. (2017). Deep counterfactual networks with propensity-dropout. *arXiv preprint arXiv:1706.05966*.

Almond, D., Chay, K. Y., & Lee, D. S. (2005). The costs of low birth weight. *The Quarterly Journal of Economics*, *120*(3), 1031–1083.

Chipman, H. A., George, E. I., & McCulloch, R. E. (2012). Bart: Bayesian additive regression trees. *Annals of Applied Statistics*, *6*(1), 266–298.

Crump, R. K., Hotz, V. J., Imbens, G. W., & Mitnik, O. A. (2008). Nonparametric tests for treatment effect heterogeneity. *The Review of Economics and Statistics*, *90*(3), 389–405.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.

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

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

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., & Winther, O. (2016). Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning* (pp. 1558–1566).

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling,

decisions. *Journal of the American Statistical Association*, *100*(469), 322–331.

Shalit, U., Johansson, F. D., & Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning* (pp. 3076–3085).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929–1958.

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

Yoon, J., Jordon, J., & Van Der Schaar, M. (2018). Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International conference on learning representations.*

Table 1

Comparison of the results among models

Method	$\sqrt{\hat{\epsilon}_{PEHE}}$		$\hat{\epsilon}_{ATE}$	
	In-sample	Out-sample	In-sample	Out-sample
GANITE	**.315** ± .006	.333 ± .013	.016 ± .006	.014 ± .014
OLS	.341 ± .005	.341 ± .007	.005 ± .004	**.007** ± .006
k-NN	.319 ± .001	**.319** ± .006	.016 ± .001	.016 ± .006
BART	.341 ± .006	.348 ± .012	**.002** ± .006	.009 ± .013
Original GANITE	.289 ± .005	.297 ± .016	.016 ± .006	.009 ± .001
Original OLS	.319 ± .001	.318 ± .007	.004 ± .003	.007 ± .006
Original k-NN	.333 ± .001	.345 ± .007	.003 ± .002	.005 ± .004
Original BART	.347 ± .009	.338 ± .016	.121 ± .024	.127 ± .023

Note that **bold** is the best estimator within our experiments in each column.

Table 2

Comparison of the results for ablation study

GANITE	$\sqrt{\hat{\epsilon}_{PEHE}}$		$\hat{\epsilon}_{ATE}$	
	In-sample	**Out-sample**	**In-sample**	**Out-sample**
Original	**.315** ± .006	.333 ± .013	.016 ± .006	.014 ± .014
AdamW	.332 ± .006	.330 ± .012	**.010** ± .007	**.001** ± .013
Dropput	.382 ± .006	.382 ± .012	.011 ± .007	.016 ± .013
Deep/AW/SL	.321 ± .006	.315 ± .012	.014 ± .007	.021 ± .013
Deep/DP/AW/SL	.326 ± .006	**.297** ± .012	.016 ± .007	.010 ± .012
Original GANITE	.289 ± .005	.297 ± .016	.016 ± .006	.009 ± .001

Note that **bold** is the best estimator in each column.

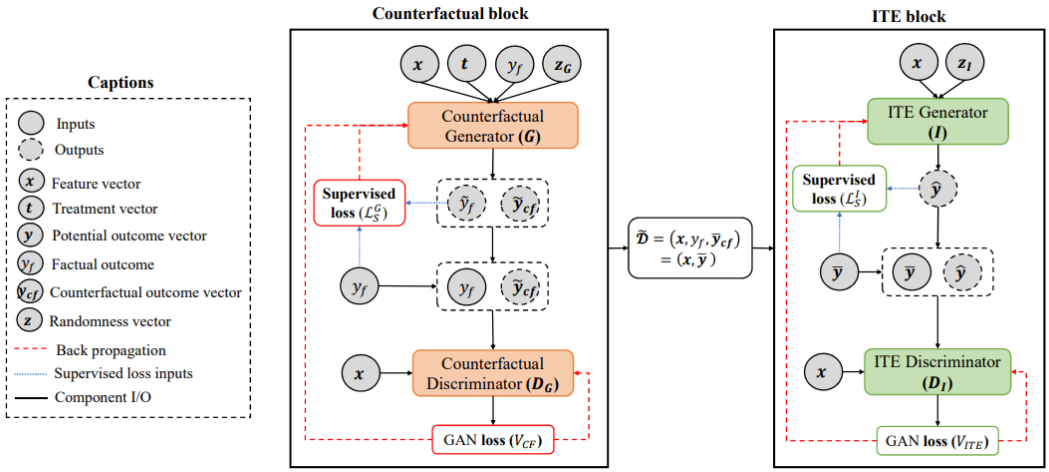

Figure 1. Block Diagram of GANITE from the original paper (Yoon et al. (2018))