Variational Causal Inference

Anonymous Author(s)
Affiliation
Address
email

Abstract

Estimating an individual's potential outcomes under counterfactual treatments is a challenging task for traditional causal inference and supervised learning approaches when the outcome is high-dimensional (e.g. gene expressions, impulse responses, human faces) and covariates are relatively limited. In this case, to construct one's outcome under a counterfactual treatment, it is crucial to leverage individual information contained in its observed factual outcome on top of the covariates. We propose a deep variational Bayesian framework that rigorously integrates two main sources of information for outcome construction under a counterfactual treatment: one source is the individual features embedded in the high-dimensional factual outcome; the other source is the response distribution of similar subjects (subjects with the same covariates) that factually received this treatment of interest.

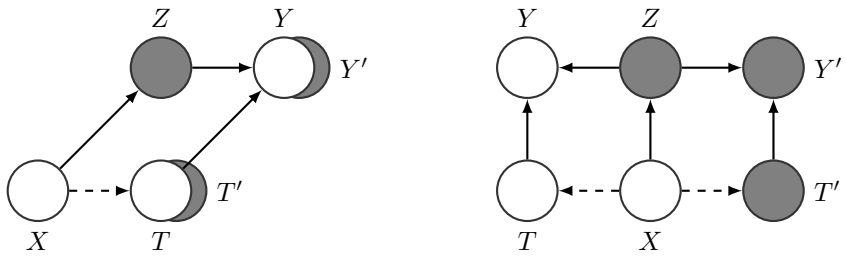
1 Introduction

In traditional causal inference, individualized treatment effect (ITE) is typically estimated by fitting supervised learning models on the covariate-specific efficacy $p(Y|X,T)$ or $\mathbb{E}[Y|X,T]$ with observed outcome Y, covariates X and treatment T. However, in cases such as the single-cell genetic perturbation datasets [2; 8; 12] where Y has thousands of dimensions while X has less than a dozen features and mostly categorical, such model could hardly be relied on to produce useful individualized results. To construct outcome Y' under a counterfactual treatment T', it is important and necessary to learn the individual-specific efficacy $p(Y'|Y,X,T,T')$, which leverages the rich information embedded in the factual outcome Y that cannot be recovered by the handful of covariates.

For example, given a cell with type A549 (X) that received SAHA drug treatment (T), we may want to know what its gene expression profile (Y') would look like if he had received Dex drug treatment (T') instead. In this case, we would want to take the profiles of other cells with type A549 that have indeed received Dex drug treatment as reference, but would also want to extract as much individual features as possible from this cell's own expression profile (Y) to combine with the reference so that the reconstructed counterfactual expression profile could preserve its own identity.

Yet the lack of observability of Y' presents a major difficulty for supervised learning when Y is taken as an input. In previous works that involve self-supervised counterfactual generators [7; 18; 6], there is no explicit regulation on the trade-off between covariate-specific efficacy and individual features unaccounted for by the covariates. In Louizos et al. [7] and Yoon et al. [18], there is no such regulation; the trade-off is black-boxed and has to depend on network sizing. In Lotfollahi et al. [6], the trade-off is implicitly encouraged by the balancing between the adversarial reward and reconstruction loss of the generator. In this work, we present a semi-autoencoding scheme and its variational Bayes [4; 11] counterpart to address this problem. Our framework is able to encode latents that explicitly honor covariate-specific efficacy while preserving individuality.

Submitted to the NeurIPS Causal ML for Real-World Impact Workshop (CML4Impact 2022). Do not distribute.



(a) folded (b) unfolded

Figure 1: The causal relation diagram. Each individual has a covariate-dependant feature state Z. Treatment T (or counterfactual treatment T') along with Z determines outcome Y (or counterfactual outcome Y'). In the causal diagram, white nodes are observed and dark grey nodes are unobserved; dashed edges exist if the data were not generated from a completely randomized trial.

2 Proposed Method

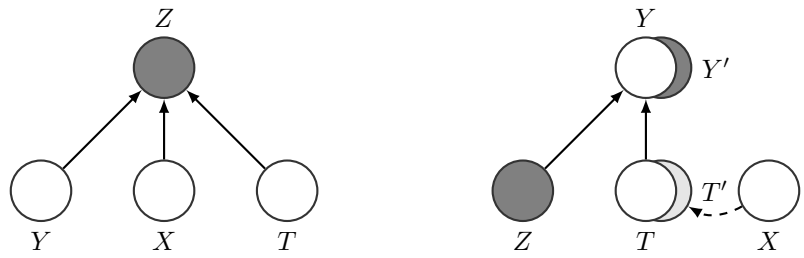
2.1 Semi-autoencoding Potential Outcomes

We define outcome $Y : \Omega \to \mathbb{R}^n$ to be a n-dimensional random vector, $X : \Omega \to E_X$ to be a mix of m categorical and real-valued covariates, $T : \Omega \to E_T$ to be a categorical or real-valued treatment and $Z : \Omega \to \mathbb{R}^d$ to be the latent feature vector on a probability space (Ω, Σ, P). Suppose the causal relations between random variables (and random vectors) follow a structural causal model (SCM) [10] depicted by the causal diagram in Figure 1, where we formulate counterfactuals Y' and T' as separate variables apart from Y and T, but having a conditional outcome distribution $p(Y'|Z, T' = a)$ identical to that of its factual counterpart $p(Y|Z, T = a)$ on a treatment level a. Y' and T' are parts of the full data $D = (X, Z, T, T', Y, Y')$, yet are never observed. Latent Z is designed to be a feature vector of Y such that random factor U_Y in structural equation $Y = f(Z, T, U_Y)$ is significantly less noisy than that of the outcome equation $Y = f(X, T, U_Y)$ in a traditional SCM of triplets (Y, X, T). Under this setting, Z can be recovered by a distribution $q(Z|Y, X, T)$ given the observations; Y and Y' can be constructed by $p(Y|Z, T)$ and $p(Y'|Z, T')$ respectively. Similar to prior works on autoencoder [17; 1], we estimate q and p with deep neural networks q_ϕ and p_θ, whose dimensionalities control the noise level of U_Y. Figure 2 shows the graphical model for the encoder and decoder.

While the reconstruction of Y is self-supervised, we can only assess the construction of Y' by looking at its resemblance to similar individuals that indeed received treatment T'. Hence we may develop a semi-autoencoding scheme that works as follows. During training, we estimate Z given the observed triplets (Y, X, T), then predict both factual and counterfactual outcomes $\tilde{Y}_{\theta,\phi}$ and $\tilde{Y}'_{\theta,\phi}$. Inputting T into decoder p_θ yields $\tilde{Y}_{\theta,\phi}$, which is then evaluated by the reconstruction loss $L(\tilde{Y}_{\theta,\phi}, Y)$; inputting T' into p_θ yields $\tilde{Y}'_{\theta,\phi}$, which we evaluate by the negative likelihood loss $-\mathcal{L}_{p(Y'|X,T')}(\tilde{Y}'_{\theta,\phi})$ with respect to the counterfactual outcome distribution $p(Y'|X, T')$. The intuition is that, if $\tilde{Y}'_{\theta,\phi}$ is indeed one's outcome under T', then the likelihood of $\tilde{Y}'_{\theta,\phi}$ coming from the outcome distribution of individuals with the same attributes that factually received treatment T' is supposed to be high. The total loss of the semi-autoencoder can then be formed as a weighted combination:

$$L(\theta, \phi) = L(\tilde{Y}_{\theta,\phi}, Y) - \omega \cdot \mathcal{L}_{\hat{p}(Y'|X,T')}(\tilde{Y}'_{\theta,\phi}) \tag{1}$$

where ω is a scaling coefficient and $\hat{p}$ is the traditional covariate-specific outcome model fit on the observed variables (X, T, Y) (notice that $p(Y'|X, T' = a) = p(Y|X, T = a)$ for any a). If $\hat{p}$ is difficult to fit, we can train a discriminator $\mathcal{D}(X, T, Y)$ for $p(Y|X, T)$ (and $p(Y'|X, T')$) instead with the adversarial approach [3] and use its discrimination loss over counterfactuals $L_{\mathcal{D}}(X, T', \tilde{Y}'_{\theta,\phi})$ in place of $-\mathcal{L}_{\hat{p}(Y'|X,T')}(\tilde{Y}'_{\theta,\phi})$. In cases where covariates are limited and discrete such as the single-cell perturbation datasets, $\hat{p}$ can simply be the empirical outcome distribution under T stratified by X.



(a) encoder (b) decoder

Figure 2: Structure of the encoder and decoder. White nodes are observed, light grey node is assigned and dark grey nodes are inferred; dashed edge is optional. Note that the decoder estimates the conditional outcome distribution of Y', in which case T' need not necessarily be sampled according to a certain true distribution $p(T'|X)$ during optimization.

2.2 Variational Causal Inference

Now we present our main result by rigorously formulating the objective, providing a probabilistic theoretical backing, and specifying an optimization scheme that reflects the intuition described in the previous section. Suppose we want to optimize $p(Y'|Y, X, T, T')$ in addition to the traditional outcome model $p(Y|X, T)$. The following theorem states the evidence lower bound (ELBO) and thus provides a roadmap for stochastic optimization:

Theorem 1. *Suppose random vector $D = (X, Z, T, T', Y, Y')$ follows a causal structure defined by the Bayesian network in Figure 1. Then $\log[p(Y'|Y, X, T, T')] + \log[p(Y|X, T)]$ has the following evidence lower bound:*

$$\log[p(Y'|Y, X, T, T')] + \log[p(Y|X, T)] \geq \mathbb{E}_{q(Z|Y,X,T)} \log[p(Y|Z, T)] + \log[p(Y'|X, T')] - \mathrm{KL}[q(Z|Y, X, T) \parallel q(Z|Y', X, T')]. \quad (2)$$

Proof of the theorem can be found in Appendix A.1. Theorem 1 states that, in order to maximize $\log[p(Y'|Y, X, T, T')] + \log[p(Y|X, T)]$, we can instead estimate and maximize the ELBO which consists of the expected reconstruction likelihood $E_{q(Z|Y,X,T)} \log[p(Y|Z, T)]$ and covariate-specific outcome likelihood $\log[p(Y'|X, T')]$, echoing the intuition highlighted by Equation 1, with an additional divergence term $-\mathrm{KL}[q(Z|Y, X, T) \parallel q(Z|Y', X, T')]$ that regularizes the similarity across latent distributions under different potential scenarios. The objective after weighting and parameterizing with deep neural networks is given as

$$J(\theta, \phi) = \mathbb{E}_{q_\phi(Z|Y,X,T)} \log[p_\theta(Y|Z, T)] + \omega_1 \cdot \log\left[p(\tilde{Y}'_{\theta,\phi}|X, T')\right] - \omega_2 \cdot \mathrm{KL}\left[q_\phi(Z|Y, X, T) \parallel q_\phi(Z|\tilde{Y}'_{\theta,\phi}, X, T')\right] \quad (3)$$

where ω_1 and ω_2 are scaling coefficients; $\tilde{Y}'_{\theta,\phi} \sim E_{q_\phi(Z|Y,X,T)} p_\theta(Y'|Z, T')$ and $p(\tilde{Y}'_{\theta,\phi}|X, T')$ can be estimated by $\hat{p}(\tilde{Y}'_{\theta,\phi}|X, T')$ with the approaches described at the end of Section 2.1. Note that this framework is fundamentally different from variational autoencoder (VAE) based frameworks such as CVAE [13] and CEVAE [7] since the variational lower bound is derived directly from the causal objective and the KL-divergence term serves a causal purpose that is completely different from the latent bounding purpose of VAEs. An explanation of this divergence term is given below.

Divergence Interpretation The divergence term $-\mathrm{KL}\left[q_\phi(Z|Y, X, T) \parallel q_\phi(Z|\tilde{Y}'_{\theta,\phi}, X, T')\right]$ in Equation 3 encourages the preservation of individuality in counterfactual outcome constructions. In the autoencoding process, the latent variable Z is required to preserve individual information contained in Y beyond covariates X in order to obtain a quality reconstruction of Y. Yet the counterfactual construction $p_\theta(Y'|\tilde{Z}_\phi, T')$ could discard this individuality in Z once it realizes a counterfactual treatment T' is in the inputs, if the outcome construction was only penalized by the covariate-specific likelihood loss $-\log[\hat{p}(Y'|X, T')]$. Such behavior is regulated with the addition of the divergence term, since otherwise we would not be able to recover a latent distribution $q_\phi(Z|\tilde{Y}'_{\theta,\phi}, X, T')$ close to $q_\phi(Z|Y, X, T)$.

2.3 Marginal Effect Estimation

Upon obtaining individual level results, we introduce a robust estimator for the marginal treatment effect $\Psi(p) = \mathbb{E}_p(Y'|T' = a)$ of a treatment level a in the variational causal inference (VCI) framework. The theorem below gives the efficient influence function [16] of $\Psi(p)$ and thus provides a construction of an asymptotically efficient regular estimator for Ψ:

Theorem 2. *Suppose $D : \Omega \to E$ follows a causal structure defined by the Bayesian network in Figure 1, where the counterfactual conditional distribution $p(Y', T'|Z, X)$ is identical to that of its factual counterpart $p(Y, T|Z, X)$. Then $\Psi(p)$ has the following efficient influence function:*

$$\tilde{\psi}(p) = \frac{I(T = a)}{p(T|X)}(Y - \mathbb{E}_p\left[Y|Z, T\right]) + \mathbb{E}_p\left[Y'|Z, T' = a\right] - \Psi. \tag{4}$$

Relating to the doubly robust estimation of average treatment effect on treated (ATT) in traditional causal inference, the above result is quite straightforward given the causal structure and the assumptions of the theorem. Nevertheless, we still provide a thorough proof in Appendix A.2. In practice, we only need $p(Y'|Z, T' = a)$ to be identical to its factual counterpart $p(Y|Z, T = a)$ given a treatment level a such that they can be estimated with the same decoder p_θ. Since Y' and T' are both unobserved in the data, it does not bother to simply acknowledge that T' is sampled according to $p(T|X)$. By Theorem 2, the following estimator is asymptotically efficient under some regularity conditions [15]:

$$\hat{\Psi}_{\theta,\phi} = \frac{1}{n}\sum_{k=1}^{n}\left\{\frac{I(T_k = a)}{\hat{p}(T_k|X_k)}Y_k + \left(1 - \frac{I(T_k = a)}{\hat{p}(T_k|X_k)}\right)\left[\mathbb{E}_{p_\theta}(Y'|\tilde{Z}_{k,\phi}, T'_k = a)\right]\right\} \tag{5}$$

where (Y_k, X_k, T_k) are the observed variables of the k-th individual and $\tilde{Z}_{k,\phi} \sim q_\phi(Y_k, X_k, T_k)$; $\hat{p}$ is an estimation of the propensity score. As can be seen, $\hat{\Psi}_{\theta,\phi}$ takes a very similar form as the augmented inverse propensity weighted (AIPW) estimator in traditional causal inference.

Covariate-specific Marginal Effect The robust estimator within the VCI framework can also be employed in estimating covariate-specific marginal treatment effect $\Psi(p) = \mathbb{E}_p(Y'|X = c, T' = a)$ for a given covariate c of interest, by applying Equation 5 on the set of observations with $X = c$. Note that the regression adjustment term $\mathbb{E}_{p_\theta}(Y'|\tilde{Z}_{k,\phi}, T'_k = a)$ by our deep network design can vary across different subjects within the same covariate group, which is not true for that of the AIPW estimator in traditional causal inference.

3 Experiments

We tested our framework on single-cell perturbation datasets along with four benchmark models — CEVAE [7], GANITE [18], CPA [6] and a naive autoencoding method. In single-cell perturbation datasets, each observation is an n-dimensional vector (the outcome) storing the expression counts of n selected genes in a single cell, along with the perturbation (the treatment) this cell was subjected to and the cell covariates. We employed the publicly available sci-Plex dataset from Srivatsan et al. [14] (Sciplex) and the CRISPRa dataset from Schmidt et al. [12] (Marson) in our experiments. Two thousand most variable genes were selected from each dataset. Data with certain treatment-covariate combinations are held out as the out-of-distribution (OOD) set and the rest are split into training and validation set with a four-to-one ratio. All common hyperparameters of all models (network width, network depth, learning rate, decay rate, etc.) are set to the same as the defaults of CPA.

Out-of-Distribution Selections We randomly select a covariate category (e.g. a cell type) and hold out all cells in this category that received one of the twenty perturbations whose effects are the hardest to predict. We use these held-out data to compose the out-of-distribution (OOD) set. We computed the Euclidean distance between the pseudobulked gene expression of each perturbation against the rest of the dataset, and selected the twenty most distant ones as the hardest-to-predict perturbations. This is the same procedure carried out by Lotfollahi et al. [6].

Differentially-Expressed Genes In order to evaluate the quality of the predictions on the genes that were substantially affected by the perturbations, we select sets of 50 differentially-expressed (DE) genes associated with each perturbation and separately evaluate model performance on these genes. This is the same procedure carried out by Lotfollahi et al. [6].

3.1 Out-of-Distribution Predictions

Same as Lotfollahi et al. [6], for each perturbation of each covariate level (e.g. each cell type of each donor) in the OOD set, we compute the R^2 (coefficient of determination) of the average outcome predictions for all genes and DE genes using samples from the validation set against the true empirical average with samples from the OOD set. The average R^2 over all perturbations of all covariate levels is then calculated as the evaluation metric. Table 1 shows the best results of each model over five independent runs. All models are trained on Amazon's accelerated computing EC2 instance G4dn (12xlarge) which contains 4 NVIDIA T4 Tensor Core GPUs.

Table 1: Results on OOD sets

	Sciplex [14]		Marson [12]	
	all genes	DE genes	all genes	DE genes
AE[§]	0.785	0.443	0.824	0.458
CEVAE [7]	0.781	0.448	0.814	0.437
GANITE[¶] [18]	0.763	0.429	0.806	0.476
CPA [6]	**0.837**	0.488	0.879	0.569
VCI	**0.837**	**0.504**	**0.893**	**0.658**

In the experiments, we used the empirical outcome distribution stratified by X and T (both discrete) to estimate the covariate-specific outcome model $p(Y|X,T)$. There are a few gradient detaching options for $\tilde{Y}'_{\theta,\phi}$ when inputting $\tilde{Y}'_{\theta,\phi}$ into the KL-divergence term $\mathrm{KL}\left[q_\phi(Z|Y,X,T) \parallel q_\phi(Z|\tilde{Y}'_{\theta,\phi},X,T')\right]$: not detached; detached from encoder; detached from both encoder and decoder. Here we chose to not detach $\tilde{Y}'_{\theta,\phi}$. Our variational Bayesian causal inference framework beat all state-of-the-art models in these experiments, with the largest fractional improvement on DE genes which are most causally affected by the perturbations.

3.2 Marginal Estimations

In this section, we use the same evaluation metric, but compute each R^2 with the robust marginal estimator and compare the results to that of the empirical mean estimator. Note that on OOD set, the robust estimator reduces to empirical mean since no perturbation-covariates combination exist in validation set. Therefore, we compute the R^2 of the marginal estimators using samples from the training set against the true empirical average on the validation set in these experiments. In each run, we train a VCI model for individualized outcome predictions, and calculate the evaluation metric for each marginal estimator periodically during the course of training. Table 2 shows the results on Marson over five independent runs. Note that the goal of robust estimation is to produce less biased estimators with tighter confidence bounds, hence here we reported the mean and standard deviation across runs instead of the maximum to reflect its effectiveness regarding this goal.

Table 2: Comparison of marginal estimators on Marson [12]

	All Genes		DE Genes	
Episode	mean	robust	mean	robust
40	0.9141 ± 0.0159	$\mathbf{0.9343 \pm 0.0080}$	0.7108 ± 0.0735	$\mathbf{0.9146 \pm 0.0305}$
80	0.9171 ± 0.0104	$\mathbf{0.9349 \pm 0.0068}$	0.7274 ± 0.0462	$\mathbf{0.9163 \pm 0.0254}$
120	0.9204 ± 0.0097	$\mathbf{0.9352 \pm 0.0063}$	0.7526 ± 0.0447	$\mathbf{0.9182 \pm 0.0229}$
160	$0.9157 \pm \mathbf{0.0043}$	$\mathbf{0.9355} \pm 0.0053$	0.7383 ± 0.0402	$\mathbf{0.9203 \pm 0.0199}$

As is shown in the table, the robust estimator provides a crucial adjustment to the empirical mean of model predictions especially on the hard-to-predict elements of high-dimensional vectors. Such estimation could be valuable in many contexts involving high-dimensional predictions where deep learning models might plateau at rather low ceilings.

[§] A naive approach adapting Autoencoder to counterfactual generation. See Appendix B.1.

[¶] GANITE's counterfactual block. GANITE's counterfacutal generator does not scale with a combination of high-dimensional outcome and multi-level treatment, see Appendix B.2 for the adaptation we made.

4 Conclusion

In this work, we introduced a variational Bayesian causal inference framework for high-dimensional individualized treatment effect prediction. With this framework, covariate-specific efficacy and individual identity can be explicitly balanced and optimized. In experiments, our framework excelled the state-of-the-art algorithms at out-of-distribution predictions on single-cell perturbation datasets — a widely accepted and biologically meaningful task. We note that although several approaches were described for the estimation of covariate-specific distribution loss, our most compelling method is for discrete treatments, where the distribution can be estimated empirically via stratification.

References

[1] Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In *International Conference on Machine Learning*, pp. 226–234. PMLR, 2014.

[2] Atray Dixit, Oren Parnas, Biyu Li, Jenny Chen, Charles P Fulco, Livnat Jerby-Arnon, Nemanja D Marjanovic, Danielle Dionne, Tyler Burks, Raktima Raychowdhury, et al. Perturb-seq: dissecting molecular circuits with scalable single-cell rna profiling of pooled genetic screens. *cell*, 167(7):1853–1866, 2016.

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[4] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[5] Jonathan Levy. Tutorial: Deriving the efficient influence curve for large models. *arXiv preprint arXiv:1903.01706*, 2019.

[6] Mohammad Lotfollahi, Anna Klimovskaia Susmelj, Carlo De Donno, Yuge Ji, Ignacio L Ibarra, F Alexander Wolf, Nafissa Yakubova, Fabian J Theis, and David Lopez-Paz. Learning interpretable cellular responses to complex perturbations in high-throughput screens. *bioRxiv*, 2021.

[7] Christos Louizos, Uri Shalit, Joris M Mooij, David Sontag, Richard Zemel, and Max Welling. Causal effect inference with deep latent-variable models. *Advances in neural information processing systems*, 30, 2017.

[8] Thomas M Norman, Max A Horlbeck, Joseph M Replogle, Alex Y Ge, Albert Xu, Marco Jost, Luke A Gilbert, and Jonathan S Weissman. Exploring genetic interaction manifolds constructed from rich single-cell phenotypes. *Science*, 365(6455):786–793, 2019.

[9] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

[10] Judea Pearl. Causal diagrams for empirical research. *Biometrika*, 82(4):669–688, 1995.

[11] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

[12] Ralf Schmidt, Zachary Steinhart, Madeline Layeghi, Jacob W Freimer, Raymund Bueno, Vinh Q Nguyen, Franziska Blaeschke, Chun Jimmie Ye, and Alexander Marson. Crispr activation and interference screens decode stimulation responses in primary human t cells. *Science*, 375(6580): eabj4008, 2022.

[13] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[14] Sanjay R Srivatsan, José L McFaline-Figueroa, Vijay Ramani, Lauren Saunders, Junyue Cao, Jonathan Packer, Hannah A Pliner, Dana L Jackson, Riza M Daza, Lena Christiansen, et al. Massively multiplex chemical transcriptomics at single-cell resolution. *Science*, 367(6473): 45–51, 2020.

[15] Mark J Van Der Laan and Daniel Rubin. Targeted maximum likelihood learning. *The international journal of biostatistics*, 2(1), 2006.

[16] Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

[17] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

[18] Jinsung Yoon, James Jordon, and Mihaela Van Der Schaar. Ganite: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section 3.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 4.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] To the best of our knowledge, there are no negative social impacts associated with our work.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes] See Section A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] `https://osf.io/ka7jh/?view_only=d265d164439b4aaab42cbca1f3ef20f0`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See Table 1 and Table 2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 3.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] See Section 3.
(b) Did you mention the license of the assets? [Yes] See Section 3.
(c) Did you include any new assets either in the supplemental material or as a URL? [No]
(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No personal identifiable information or offensive content is presented on this manuscript

5. If you used crowdsourcing or conducted research with human subjects...
(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Appendix

A Proof of Theorems

A.1 Proof of Theorem 1

Proof. By the d-separation [9] of paths on the causal graph defined in Figure 1, we have

$$\begin{aligned}
\log\left[p(Y'|Y,X,T,T')\right] &= \log \mathbb{E}_{q(Z|Y,X,T)}\left[p(Y'|Z,Y,X,T,T')\right] && (6)\\
&\geq \mathbb{E}_{q(Z|Y,X,T)} \log\left[p(Y'|Z,Y,X,T,T')\right] \quad \text{(Jensen's inequality)} && (7)\\
&= \mathbb{E}_{q(Z|Y,X,T)} \log \frac{p(Y',Z|Y,X,T,T')}{q(Z|Y,X,T)} && (8)\\
&= \mathbb{E}_{q(Z|Y,X,T)} \log \frac{p(Y',Z,Y,T|X,T')}{q(Z|Y,X,T)p(Y,T|X)} && (9)\\
&= \mathbb{E}_{q(Z|Y,X,T)} \log \frac{p(Y'|X,T')q(Z|Y',X,T')p(Y,T|Z,X)}{q(Z|Y,X,T)p(Y,T|X)} && (10)\\
&= \mathbb{E}_{q(Z|Y,X,T)} \log \frac{p(Y'|X,T')q(Z|Y',X,T')p(Y|Z,T)p(T|X)}{q(Z|Y,X,T)p(Y|X,T)p(T|X)} && (11)\\
&= \log\left[p(Y'|X,T')\right] - \mathrm{KL}\left[q(Z|Y,X,T) \parallel q(Z|Y',X,T')\right]\\
&\quad + \mathbb{E}_{q(Z|Y,X,T)} \log\left[p(Y|Z,T)\right] - \log\left[p(Y|X,T)\right] && (12)
\end{aligned}$$

Reorganizing the terms yields

$$\begin{aligned}
\log\left[p(Y'|Y,X,T,T')\right] + \log\left[p(Y|X,T)\right] \geq\ & \mathbb{E}_{q(Z|Y,X,T)} \log\left[p(Y|Z,T)\right] + \log\left[p(Y'|X,T')\right]\\
& - \mathrm{KL}\left[q(Z|Y,X,T) \parallel q(Z|Y',X,T')\right]. \qquad (13)
\end{aligned}$$

□

A.2 Proof of Theorem 2

Proof. Following Van der Vaart [16], we define a path $p_\epsilon(D) = p(D)(1+\epsilon S(D))$ on density p of D as a submodel that passes through p at $\epsilon = 0$ in the direction of the score $S(D) = \frac{d}{d\epsilon}\log\left[p_\epsilon(D)\right]\Big|_{\epsilon=0}$. Following the key identity presented in Levy [5]:

$$\begin{aligned}
&\frac{d}{d\epsilon} p_\epsilon(d_i|pa(D_i) = \bar{d}_{i-1})\Big|_{\epsilon=0}\\
&\quad = p(d_i|pa(D_i) = \bar{d}_{i-1})(\mathbb{E}[S(D)|D_i = d_i, pa(D_i) = \bar{d}_{i-1}] - \mathbb{E}[S(D)|pa(D_i) = \bar{d}_{i-1}]) \qquad (14)
\end{aligned}$$

where $pa(D_i)$ denotes the parent nodes of variable D_i and minuscule of a variable denotes the value it takes, we have

$$
\begin{aligned}
\frac{d}{d\epsilon}\Psi(p_\epsilon)\Big|_{\epsilon=0} &= \frac{d}{d\epsilon}\Big|_{\epsilon=0}\mathbb{E}_{p_\epsilon}\left[\mathbb{E}_{p_\epsilon}\left[Y'|Z,T'=a\right]\right] && (15)\\
&= \frac{d}{d\epsilon}\Big|_{\epsilon=0}\int_{y',z,x} y'\left[p_\epsilon(y'|z,T'=a)p_\epsilon(z,x)\right] && (16)\\
&= \int_{y',z,x} y'\frac{d}{d\epsilon}\Big|_{\epsilon=0}\left[p_\epsilon(y'|z,T'=a)p_\epsilon(z,x)\right] \quad \text{(dominated convergence)} && (17)\\
&= \int_{y',z,x} y'p(z,x)\frac{d}{d\epsilon}\Big|_{\epsilon=0}p_\epsilon(y'|z,T'=a) && (18)\\
&+ \int_{y',z,x} y'p(y'|z,T'=a)\frac{d}{d\epsilon}\Big|_{\epsilon=0}p_\epsilon(z,x) && (19)\\
&= \int_d I(t'=a)\frac{p(t'|x)}{p(t'|x)}y'p(y,t|z,x)p(z,x)\frac{d}{d\epsilon}\Big|_{\epsilon=0}p_\epsilon(y'|z,t')\\
&\quad + \int_{y',z,x} y'p(y'|z,T'=a)\frac{d}{d\epsilon}\Big|_{\epsilon=0}p_\epsilon(z,x) && (20)\\
&= \int_d \frac{I(t'=a)}{p(t'|x)}y'p(y',t'|z,x)p(y,t|z,x)p(z,x)\left\{S(d)-\mathbb{E}\left[S(D)|y,z,x,t,t'\right]\right\}\\
&\quad + \int_{y',z,x} y'p(y'|z,T'=a)p(z,x)\left\{\mathbb{E}\left[S(D)|z,x\right]-\mathbb{E}\left[S(D)\right]\right\} && (21)\\
&= \int_d \frac{I(t=a)}{p(t|x)}yp(y,t|z,x)p(y',t'|z,x)p(z,x)\left\{S(d)-\mathbb{E}\left[S(D)|y',z,x,t',t\right]\right\}\\
&\quad + \int_{y',z,x} y'p(y'|z,T'=a)p(z,x)\left\{\mathbb{E}\left[S(D)|z,x\right]-\mathbb{E}\left[S(D)\right]\right\} && (22)\\
&= \int_d S(d)\cdot\frac{I(t=a)}{p(t|x)}yp(d)\\
&\quad - \int_d \mathbb{E}\left[S(D)|y',z,x,t,t'\right]p(y',z,x,t,t')\cdot\frac{I(t=a)}{p(t|x)}yp(y|z,t)\\
&\quad + \int_{y',z,x}\mathbb{E}\left[S(D)|z,x\right]p(z,x)\cdot y'p(y'|z,T'=a)\\
&\quad - \int_{y',z,x}\mathbb{E}\left[S(D)\right]\cdot y'p(y'|z,T'=a)p(z,x) && (23)\\
&= \int_d S(d)\left\{\frac{I(t=a)}{p(t|x)}(y-\mathbb{E}\left[Y|z,t\right])+\mathbb{E}\left[Y'|z,T'=a\right]-\Psi\right\}p(d) && (24)
\end{aligned}
$$

by assumptions of Theorem 2 and factorization according to Figure 1. Hence

$$
\frac{d}{d\epsilon}\Psi(p_\epsilon)\Big|_{\epsilon=0} = \left\langle S(D), \frac{I(T=a)}{p(T|X)}(Y-\mathbb{E}_p\left[Y|Z,T\right])+\mathbb{E}_p\left[Y'|Z,T'=a\right]-\Psi\right\rangle_{L^2(\Omega;E)} \tag{25}
$$

and we have $\tilde{\psi}_p = I(T=a)/p(T|X)\cdot(Y-\mathbb{E}_p\left[Y|Z,T\right])+\mathbb{E}_p\left[Y'|Z,T'=a\right]-\Psi$. □

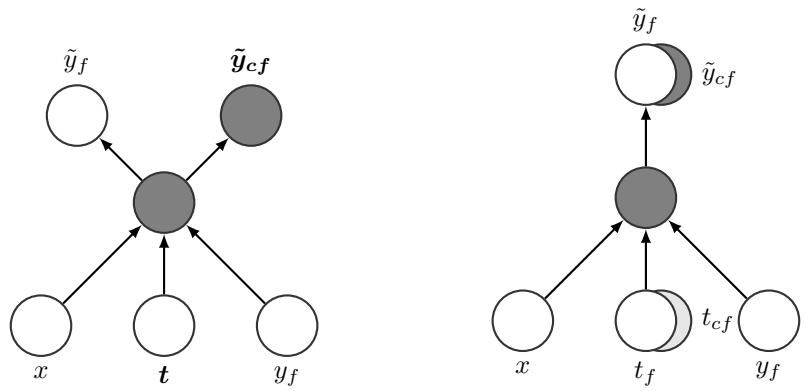
B Benchmark Adaptations

B.1 Autoencoder

The adapted autoencoder reconstructs the outcome during training similar to a generic autoencoder, but takes treatment and covariates as additional inputs. During test time, we simply plug in the counterfactual treatments along with factual outcomes and covariates to generate the counterfactual outcome predictions.

B.2 GANITE

GANITE [18]'s counterfacutal generator does not scale with a combination of high-dimensional outcome and multi-level treatment, thus here we only input one randomly sampled counterfactual treatment to the generator and correspondingly construct one counterfactual outcome for each sample. See Figure 3 for the original and adapted structure of the model.



(a) original (b) adapted

Figure 3: GANITE's counterfactual generator. t_{cf} is a random sample of $\boldsymbol{t}$, passed into the generator as a part of the input (x, t_{cf}, y_f), separately from input (x, t_f, y_f) of the factual generation.

The discriminator predicts the logits l_f, l_{cf} of y_f, $\tilde{y}_{cf}$ separately. The cross entropy loss of (l_f, l_{cf}) against $(1, 0)$ is then calculated.