Counterfactual Generative Models for Time-Varying Treatments

Shenghao Wu[1] **Wenbin Zhou**[1] **Minshuo Chen**[2] **Shixiang Zhu**[1]
[1]Carnegie Mellon University; [2]Princeton University
{shenghaw, wenbinz2, shixianz }@andrew.cmu.edu
minshuochen@princeton.edu

Abstract

Estimating the counterfactual outcome of treatment is essential for decision-making in public health and clinical science, among others. Often, treatments are administered in a sequential, time-varying manner, leading to an exponentially increased number of possible counterfactual outcomes. Furthermore, in modern applications, the outcomes are high-dimensional and conventional average treatment effect estimation fails to capture disparities in individuals. To tackle these challenges, we propose a novel conditional generative framework capable of producing counterfactual samples under time-varying treatment, without the need for explicit density estimation. Our method carefully addresses the distribution mismatch between the observed and counterfactual distributions via a loss function based on inverse probability weighting. We present a thorough evaluation of our method using both synthetic and real-world data. Our results demonstrate that our method is capable of generating high-quality counterfactual samples and outperforms the state-of-the-art baselines.

1 Introduction

Estimating time-varying treatment effect from observational data has garnered significant attention due to the growing prevalence of time-series records. One particular relevant field is public health [31, 76, 8], where researchers and policymakers grapple with a series of decisions on preemptive measures to control epidemic outbreaks, ranging from mask mandates to shutdowns. It is vital to provide accurate and comprehensive outcome estimates under such diverse time-varying treatments, so that policymakers and researchers can accumulate sufficient knowledge and make well-informed decisions with discretion.

In the literature, average treatment effect estimation has received extensive attention and various methods have been proposed [58, 23, 26, 34, 6, 4, 65, 39, 17, 71]. By estimating the average outcome over a population that receives a treatment or policy of interest, these methods evaluate the effectiveness of the treatment via hypothesis testing. However, solely relying on the average treatment effect might not capture the full picture, as it may overlook pronounced disparities in the individual outcomes of the population, especially when the counterfactual distribution is heterogeneous.

Recent efforts [29, 28, 40] have been made to directly estimate the counterfactual density function of the outcome. This idea has demonstrated appealing performance for univariate outcomes. Nonetheless, for multi-dimensional outcomes, the estimation accuracy quickly degrades [64]. In modern high-dimensional applications, for example, predicting COVID-19 cases at the county level of a state, these methods are hardly scalable and incur a computational overhead.

Adding another layer of complexity, considering time-varying treatments causes the capacity of the potential treatment sequences to expand exponentially. For example, even if the treatment is binary at a single time step, the total number of different combinations on a time-varying treatment increases as 2^d with d being the length of history. More importantly, time-varying treatments lead to significant distributional discrepancy between the observed and counterfactual outcomes, as shown in Figure 1.

Accepted to the NeurIPS 2023 Workshop on Causal Representation Learning.

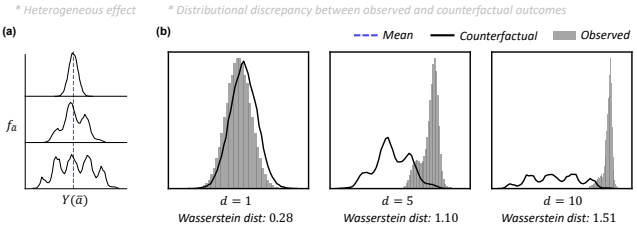


Figure 1: **(a)** Mean is incapable of describing the heterogeneous effect in counterfactual distributions. **(b)** The observed distribution may be more deviated from that of the counterfactual as the length of the history dependence, d, increases.

In this paper, we provide a whole package of accurately estimating high-dimensional counterfactual distributions for time-varying treatments. Instead of a direct density estimation, we implicitly learn the counterfactual distribution by training a generative model, capable of generating credible samples of the counterfactual outcomes given a time-varying treatment. This allows policymakers to assess a policy's efficacy by exploring a range of probable outcomes and deepening their understanding of its counterfactual result. Here, we summarize the benefits of our proposed method:

1. Our model is capable of handling high-dimensional outcomes;
2. Our model outperforms existing state-of-the-art baselines in terms of estimation accuracy and generating high-quality counterfactual samples;
3. Our model enables fast downstream inference, such as average treatment effect estimation and uncertainty quantification.

To be specific, we develop a conditional generator [41, 67]. This generator, which we choose in a flexible manner, takes into account the treatment history as input and generates counterfactual outcomes that align with the underlying distribution of counterfactuals. The key idea behind the scenes is to utilize a "proxy" conditional distribution as an approximation of the true counterfactual distribution. To achieve this, we establish a statistical relationship between the observed and counterfactual distributions inspired by the g-formula [45, 60, 54, 15]. We learn the conditional generator by optimizing a novel weighted loss function based on a pseudo population through Inverse Probability of Treatment Weighting (IPTW) [54]. We evaluate our framework through numerical experiments extensively on both synthetic and real-world data sets. A comprehensive overview of the related work is included in Appendix A.

2 Methodology

2.1 Problem setup

In this study, we consider the treatment for each discrete time period (such as day or week) as a random variable $A_t \in \mathcal{A} = \{0, 1\}$, where $t = 1, \ldots, T$ and T is the total number of time points. Note that our framework also works with categorical and even continuous treatments. Let $X_t \in \mathcal{X} \subset \mathbb{R}^h$ be the time-varying covariates, and $Y_t \in \mathcal{Y} \subset \mathbb{R}^m$ the subject's outcome at time t. We use $\overline{A}_t = \{A_{t-d+1}, \ldots, A_t\}$ to denote the previous treatment history from time $t - d + 1$ to t, where d is the length of history dependence. Similarly, we use $\overline{X}_t = \{X_{t-d+1}, \ldots, X_t\}$ to denote the covariate history. We use y_t, a_t, and x_t to represent a realization of Y_t, A_t, and X_t, respectively, and use $\overline{a}_t = (a_{t-d+1}, \ldots, a_t)$ and $\overline{x}_t = (x_{t-d+1}, \ldots, x_t)$ to denote the history of treatment and covariate realizations. In the sections below, we will refer to Y_t, $\overline{A}_t$, and $\overline{X}_t$ as simply Y, $\overline{A}$, and $\overline{X}$, where t will be clear from context.

The goal of our study is to obtain realistic samples of the counterfactual outcome given a time-varying treatment $\overline{a}$, without estimating its counterfactual density. Let $Y(\overline{a})$ denote the counterfactual outcome for a subject under a time-varying treatment $\overline{a}$, and define $f_{\overline{a}}$ as its counterfactual distribution. We note that $f_{\overline{a}}$ is different from the marginal density of Y, as the treatment is fixed at $\overline{A} = \overline{a}$ in the conditioning. It is also not equal to the unadjusted conditional density $f(y|\overline{a})$. Instead, $f_{\overline{a}}$ is the density of the counterfactual variable $Y(\overline{a})$, which represents the outcome that would have been observed if treatment were set to $\overline{A} = \overline{a}$, if the standard assumptions [56, 34] hold. See Appendix B for more details about assumptions. We also assume that Y, $\overline{A}$, and $\overline{X}$ follows the typically structural causal relationship as shown in Figure 7 (Appendix C), which is a classical setting in longitudinal causal inference [51, 56].

2.2 Counterfactual generative framework for time-varying treatments

This paper proposes a counterfactual generator, denoted as g_θ, to simulate $Y(\overline{a})$ according to the *proxy conditional distribution* $f_\theta(y|\overline{a})$ instead of directly modeling its expectation or specifying a parametric counterfactual distribution. Here we use $\theta \in \Theta$ to represent the model's parameters, and formally define the generator as a function:

$$g_\theta(z, \overline{a}) : \mathbb{R}^r \times \mathcal{A}^d \to \mathcal{Y}. \tag{1}$$

The generator takes as input a random noise vector ($z \in \mathbb{R}^r \sim \mathcal{N}(0, I)$) and the time-varying treatment $\overline{a}$.

The output of the generator is a sample of possible counterfactual outcomes that follows the proxy conditional distribution represented by θ, *i.e.*,

$$y \sim f_\theta(\cdot|\overline{a}),$$

which can be viewed as an approximate of the underlying counterfactual distribution $f_{\overline{a}}$. Figure 2 shows an overview of the proposed generative model architecture.

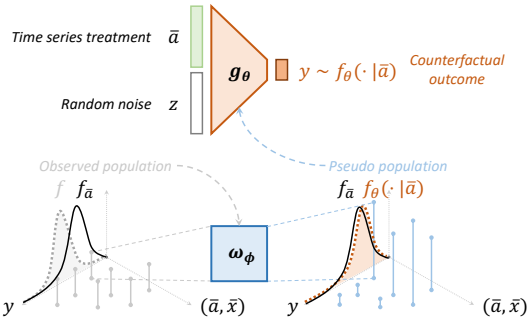


Figure 2: The architecture of the proposed counterfactual generative models. The generator g_θ is designed to produce samples of the outcome variable $Y(\overline{a})$ with a given time-varying treatment $\overline{a}$. The generated samples are expected to conform to the proxy conditional distribution f_θ, which is an approximate of the underlying counterfactual distribution $f_{\overline{a}}$.

The learning objective is to find the optimal generator that minimizes the distance between the proxy conditional distribution $f_\theta(\cdot|\overline{a})$ and the true counterfactual distribution $f_{\overline{a}}$, as illustrated in Figure 3. If the distance metric is Kullback-Leibler (KL) divergence, this objective can be expressed equivalently by maximizing the log-likelihood [43]:

$$\max_{\theta \in \Theta} \ell(\theta) := \mathop{\mathbb{E}}_{y \sim f_{\overline{a}}} \log f_\theta(y|\overline{a}). \tag{2}$$

To obtain samples from the counterfactual distribution $f_{\overline{a}}$, we follow the idea of marginal structural models (MSMs) introduced by [45, 60, 54] and extended by [15] to account for time-varying treatments. Specifically, we introduce Lemma 1, which follows the g-formula proposed in [54] and establishes a connection between the counterfactual distribution and the data distribution. The proof can be found in Appendix C.

Lemma 1. *Under unconfoundedness and positivity, we have*

$$f_{\overline{a}}(y) = \int \frac{\mathbb{1}\{\overline{A} = \overline{a}\}}{\prod_{\tau=t-d}^{t} f\left(A_\tau | \overline{A}_{\tau-1}, \overline{X}_\tau\right)} f\left(y, \overline{A}, \overline{X}\right) d\overline{A} d\overline{X}, \tag{3}$$

where f denotes the observed data distribution.

Now we present a proposition using Lemma 1, allowing us to substitute the expectation in (2), computed over a counterfactual distribution, with the sample average over a pseudo-population. This pseudo-population is constructed by assigning weights to each data tuple based on their subject-specific IPTW. Figure 3 gives an illustration of the learning objective. See the proof in Appendix D.

Proposition 1. *Let $\mathcal{D}$ denote the set of observed data tuples. The generative learning objective can be approximated by:*

$$\mathop{\mathbb{E}}_{y \sim f_{\overline{a}}} \log f_\theta(y|\overline{a}) \approx \sum_{(y, \overline{a}, \overline{x}) \in \mathcal{D}} w_\phi(\overline{a}, \overline{x}) \log f_\theta(y|\overline{a}), \tag{4}$$

where $w_\phi(\overline{a}, \overline{x})$ denotes the subject-specific IPTW, parameterized by $\phi \in \Phi$, which takes the form:

$$w_\phi(\overline{a}, \overline{x}) = \frac{1}{\prod_{\tau=t-d}^{t} f_\phi(a_\tau | \overline{a}_{\tau-1}, \overline{x}_\tau)}. \tag{5}$$

Remark 1. *The generative learning objective in Proposition 1 offers specific benefits when compared to plugin methods using Lemma 1 [7, 28] and parametric density estimation [40]. The use of IPTW over doubly robust methods can be supported due to its computational efficiency and the potentially lower chance of model misspecification in high-dimensional outcome settings. We include a detailed discussion in Appendix E.*

Here we use another model, denoted by $\phi \in \Phi$, to represent the conditional probability $f(A_\tau|\overline{A}_\tau, \overline{X}_\tau)$, which defines the IPTW w_ϕ and can be learned separately using observed data. Note that the effectiveness of this method is dependent on a correct specification of the IPTW ϕ, *i.e.*, the black dot is inside of the blue area in Figure 3 [15, 34]. In [34], they use an RNN-like structure to represent the conditional probability f_ϕ without making strong assumptions on the form of the conditional probability. The choice of g_θ and f_ϕ are entirely general and both can be specified using deep architectures. In this paper, we use fully-connected neural networks for both g_θ and f_ϕ.

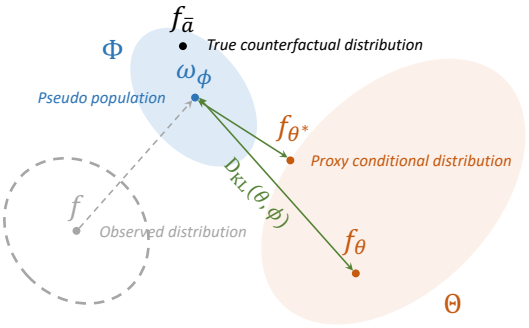


Figure 3: An illustration of our learning objective. We aim to minimize the KL-divergence between the proxy conditional distribution $f_\theta(\cdot|\bar{a})$ and the true counterfactual distribution $f_{\bar{a}}$.

Variational approximation and learning To compute the weighted log-likelihood as expressed in (4) and learn the proposed generative model, we can leverage various state-of-the-art generative learning algorithms, such as conditional normalizing flow [5] and guided diffusion models [13]. In this paper, we adopt the conditional variational autoencoder (CVAE) [67], a commonly-used learning algorithm for generative models, approximate the logarithm of the proxy conditional probability using its evidence lower bound (ELBO):

$$\log f_\theta(y|\bar{a}) \geq -D_{\mathrm{KL}}\left(q(z|y,\bar{a})||p_\theta(z|\bar{a})\right) + \mathbb{E}_{q(z|y,\bar{a})}\left[\log p_\theta(y|z,\bar{a})\right], \tag{6}$$

where q is a variational approximation of the posterior distribution over the random noise given observed outcome y and its treatment $\bar{a}$. The first term on the right-hand side is the Kullback–Leibler (KL) divergence of the approximate posterior $q(\cdot|y,\bar{a})$ from the exact posterior $p_\theta(\cdot|\bar{a})$. The second term is the log-likelihood of the latent data-generating process. The complete derivation of (6) and implementation details can be found in Appendix F. We summarize our learning procedure in Algorithm 1.

3 Experiments

We evaluate our method using numerical examples and demonstrate the superior performance compared to five state-of-the-art methods. These are (1) Kernel Density Estimation (`KDE`) [59], (2) Marginal structural model with a fully-connected neural network (`MSM+NN`) [55, 34], (3) Conditional Variational Autoencoder (`CVAE`) [67], (4)Semi-parametric Plug-in method based on pseudo-population (`Plugin+KDE`) [29], and (5) G-Net (`G-Net`) [33]. In the following, we refer to our proposed conditional event generator as marginal structural conditional variational autoencoder (`MSCVAE`). See Appendix G.1 for a detailed review of these baseline methods. More details of the evaluation metrics and experiment set-up can be found in Appendix G.2 and G.3.

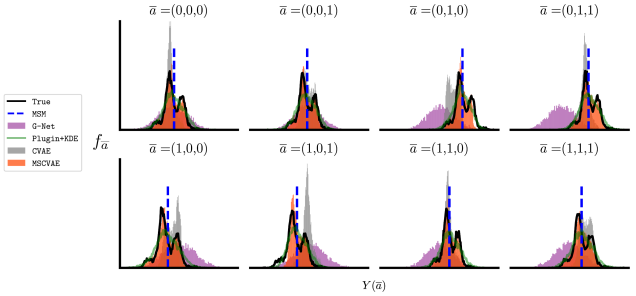


Figure 4: The estimated and true counterfactual distributions for $(d = 3)$ on the fully synthetic datasets $(m = 1)$. We include the plot for $d = 1$ and $d = 5$ in Appendix H.

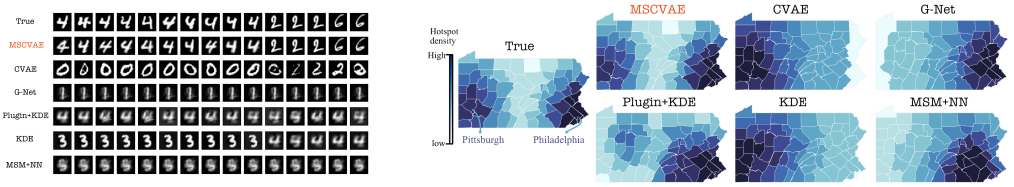
3.1 Fully synthetic data

We first assess the effectiveness of the `MSCVAE` using fully synthetic experiments. Following the classical experimental setting described in [55], we simulate three synthetic datasets with different lengths of history dependence $(d = 1, 3, 5)$ using linear models. See Appendix G.4 for a detailed description of the synthetic data generation. It is worth mentioning that the learned distribution produced by `CVAE` deviates significantly from the desired target, emphasizing the significance of the weighting term in Proposition 1 in accurately approximating the counterfactual distribution. Table 1 summarizes the quantitative comparisons across the baselines. The `MSCVAE` not only consistently achieves the smallest Wasserstein distance in the majority of the experimental settings,

Table 1: Quantitative performance on fully-synthetic and semi-synthetic data

	Fully synthetic ($m = 1$)						**COVID-19**	**TV-MNIST**
	$d = 1$		$d = 3$		$d = 5$		$m = 67$	$m = 784$
Methods	Mean ↓	Wasserstein ↓	Mean ↓	Wasserstein ↓	Mean ↓	Wasserstein ↓	FID* ↓	FID* ↓
`MSM+NN`	**0.001** (**0.002**)	0.601 (0.603)	0.070 (**0.159**)	0.689 (0.718)	0.198 (**0.563**)	0.600 (0.737)	1.085 (1.665)	**1.236** (3.956)
`KDE`	0.246 (0.267)	0.244 (0.268)	0.520 (1.080)	0.538 (1.080)	0.538 (1.419)	0.539 (1.419)	0.981 (2.665)	1.509 (2.557)
`Plugin+KDE`	0.010 (0.014)	**0.034** (**0.036**)	**0.045** (0.168)	**0.132** (**0.168**)	**0.147** (0.598)	**0.182** (**0.598**)	0.652 (**0.759**)	1.370 (**1.799**)
`G-Net`	0.211 (0.258)	0.572 (0.582)	1.167 (2.173)	1.284 (2.173)	2.314 (5.263)	2.354 (5.263)	0.965 (1.856)	1.751 (6.096)
`CVAE`	0.250 (0.287)	0.253 (0.288)	0.517 (1.061)	0.553 (1.061)	0.539 (1.430)	0.613 (1.430)	**0.641** (2.654)	2.149 (5.484)
`MSCVAE`	**0.006** (**0.006**)	**0.055** (**0.056**)	**0.046** (**0.150**)	**0.105** (**0.216**)	**0.150** (**0.633**)	**0.173** (**0.633**)	**0.336** (**0.712**)	**0.270** (**1.004**)

* Numbers represent the average metric across all treatment combinations and those in the parentheses represent the worst across treatment combinations.



(a) (b)

Figure 5: **(a)** Results on the semi-synthetic TV-MNIST datasets ($m = 784$). We show representative samples generated from different methods under the treatment combinations $\overline{a} = (1, 1, 1)$. **(b)** Results on the semi-synthetic Pennsylvania COVID-19 mask datasets ($m = 67$) under the treatment combination $\overline{a} = (1, 1, 1)$. We visualize the distribution of "hotspots" from the generated and true counterfactual distribution. For each model, we generate 500 counterfactual samples. Each sample is a 67-dimensional vector representing the inferred new cases per 100K for the counties in Pennsylvania. We define the hotspot of each sample as the coordinate of the county with the highest number of new cases per 100K, and visualize the density of the 500 hotspots using kernel density estimation.

but also demonstrates highly competitive accuracy on mean estimation, which is consistent with the result in Figure 4. Note that even though our goal is not to explicitly estimate the counterfactual distribution, the results clearly demonstrate that our generative model can still accurately approximate the underlying counterfactual distribution, even compared to the unbiased density-based method such as `Plugin+KDE`.

3.2 Semi-synthetic Data

To demonstrate the ability of our generative framework to generate credible high-dimensional counterfactual samples, we test our method on two semi-synthetic datasets. The benefit of these datasets is that both factual and counterfactual outcomes are available. Therefore, we can obtain a sample from the ground-truth counterfactual distribution, which we can then use for benchmarking.

Time-varying MNIST We create TV-MNIST, a semi-synthetic dataset using MNIST images [12, 27] as the outcome variable ($m = 784$). In this dataset, images are randomly selected, driven by the result of a latent process defined by a linear autoregressive model, which takes a 1-dimensional covariate and treatment variable as inputs and outputs a digit (between 0 and 9). Here we set the length of history dependence, d, to 3. This setup allows us to evaluate the performance of the algorithms by visually contrasting the quality and distribution of generated samples against counterfactual ones. The full description of the dataset can be found in Appendix G.5.

Pennsylvania COVID-19 mask mandate We create another semi-synthetic dataset to investigate the effectiveness of mask mandates in Pennsylvania during the COVID-19 pandemic. We collected data from multiple sources, including the Centers for Disease Control and Prevention (CDC), the US Census Bureau, and a Facebook survey [78, 16, 79, 19, 9, 21]. The dataset encompasses variables aggregated on a weekly basis spanning 106 weeks from 2020 and 2022. There are four state-level covariates (per 100K people): the number of deaths, the average retail and recreation mobility, the surveyed COVID-19 symptoms, and the number of administered COVID-19 vaccine doses. We set the state-level mask mandate policy (with values of 0 indicating no mandate and 1 indicating a mandate) as the treatment variable, and the county-level number of new COVID-19 cases (per 100K) as the outcome variable ($m = 67$). We simulate $2,000$ trajectories of the (covariate, treatment) tuples of 300 time points (each point corresponding to a week) according to the real data. The outcome model is structured to exhibit a peak, defined as the "hotspot", in one of the state's two major cities: Pittsburgh or Philadelphia. The likelihood of these hotspots is contingent on the covariates. Consequently, the counterfactual and observed distributions manifest as bimodal, with varying probabilities for the

hotspot locations. To ensure a pertinent analysis window, we've fixed the history dependence length, d, at 3, aligning with the typical duration within which most COVID-19 symptoms recede [38]. The full description of the dataset can be found in Appendix G.6.

As we can observe in Figure 5a and 5b, the `MSCVAE` outperforms other baselines in generating samples that closely resembles the ground truth. This visual superiority is also reinforced by the overwhelmingly better FID* scores of `MSCVAE` compared to other baselines methods, as shown in Table 1 (see Appendix G.2 for the definition of FID*). It's worth noting that the samples produced by the `Plugin+KDE` appear blurred in Figure 5a and exhibit noise in Figure 5b. This can be attributed to the inherent complexities of high-dimensional density estimation [64]. Such observations underscore the value of employing a generative model to craft high-dimensional samples without resorting to precise density estimation. We also notice that `G-Net` fails to capture the high-dimensional counterfactual outcomes, particularly due to challenges in accurately defining the conditional outcome model and the covariate density model. The superior results of `MSCVAE` compared to `CVAE` and of `Plugin+KDE` over `KDE` emphasize the pivotal role of IPTW correction during modeling. Moreover, deterministic approaches like `MSM+NN` might fall short in capturing key features of the counterfactual distribution. In sum, the semi-synthetic experiments highlights the distinct benefits of our generative framework, particularly in generating high-quality counterfactual samples under time-varying treatments in a high-dimensional causal context.

3.3 Real Data

We perform a case study using a real COVID-19 mask mandate dataset across the U.S. from 2020 to 2021 spanning 49 weeks. Due to the limitation on the sample size for state-level observations, we only look at the county-level data, covering $3,219$ U.S. counties. This leads to $m = 1$. The details can be found in Appendix G.6. Figure 6 illustrates a comparative analysis of the distribution of the observed and generated outcome samples under two different scenarios: one without a mask mandate ($\overline{a} = (0, 0, 0)$) and the other with a full mask mandate ($\overline{a} = (1, 1, 1)$). In the left panel, we observe that the distributions under both policies appear remarkably similar, suggesting that the mask mandate has a limited impact on controlling the spread of the virus. In the right panel, we present counterfactual distributions estimated using our method, revealing a noticeable disparity between the mask mandate and no mask mandate scenarios. The mean of the distribution for the mask mandate is significantly lower than that of the no-mask mandate. These findings indicate that implementing a mask mandate consistently for three consecutive weeks can effectively reduce the number of future new cases. It aligns with the understanding supported by health experts' suggestions and various studies [69, 1, 22, 46, 73] regarding the effectiveness of wearing masks. Finally, it is important to note that the implementation of full mask mandates exhibits a significantly higher variance compared to the absence of a mask mandate. This implies that the impact of a mask mandate varies across different data points, specifically counties in our study.

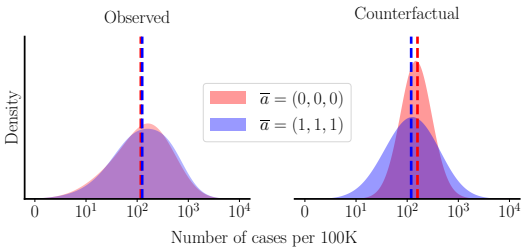


Figure 6: Observed distribution and estimated counterfactual distribution of the number of real COVID-19 cases per 100K under two mask policies. The vertical dashed lines represent the mean of the corresponding distributions.

4 Conclusions

We have introduced a powerful conditional generative framework tailored to generate samples that mirror counterfactual distributions in scenarios where treatments vary over time. Our model approximates the true counterfactual distribution by minimizing the KL-divergence between the true distribution and a proxy conditional distribution, approximated by generated samples. We have showcased our framework's superior performance against state-of-the-art methods in both fully-synthetic and real experiments.

The generative nature of our framework enables various future downstream applications including uncertainty quantification and can be enhanced by adopting cutting edge generative learning algorithms, such as diffusion models. Additionally, our generative approach can be easily adapted to scenarios with continuous treatments, where the conditional generator enables extrapolation between unseen treatments under continuity assumptions.

References

[1] Dhaval Adjodah, Karthik Dinakar, Matteo Chinazzi, Samuel P Fraiberger, Alex Pentland, Samantha Bates, Kyle Staller, Alessandro Vespignani, and Deepak L Bhatt. Association between covid-19 outcomes and mask mandates, adherence, and attitudes. *PLoS One*, 16(6):e0252315, 2021.

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

[3] Vahid Balazadeh Meresht, Vasilis Syrgkanis, and Rahul G Krishnan. Partial identification of treatment effects with implicit generative models. *Advances in Neural Information Processing Systems*, 35:22816–22829, 2022.

[4] Jeroen Berrevoets, Alicia Curth, Ioana Bica, Eoin McKinney, and Mihaela van der Schaar. Disentangled counterfactual recurrent networks for treatment effect inference over time. *arXiv preprint arXiv:2112.03811*, 2021.

[5] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction. *arXiv preprint arXiv:1908.09008*, 2019.

[6] Ioana Bica, Ahmed M Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations. *arXiv preprint arXiv:2002.04083*, 2020.

[7] Peter J Bickel and Jaimyoung Kwon. Inference for semiparametric models: some questions and an answer. *Statistica Sinica*, pages 863–886, 2001.

[8] Matteo Bonvini, Edward Kennedy, Valerie Ventura, and Larry Wasserman. Causal inference in the time of covid-19. *arXiv preprint arXiv:2103.04472*, 2021.

[9] U.S. Census Bureau. State population totals: 2020-2022. `https://www.census.gov/data/tables/time-series/demo/popest/2020s-state-total.html`, 2022. Accessed: 2022-09-15.

[10] Yehu Chen, Annamaria Prati, Jacob Montgomery, and Roman Garnett. A multi-task gaussian process model for inferring time-varying treatment effects in panel data. In *International Conference on Artificial Intelligence and Statistics*, pages 4068–4088. PMLR, 2023.

[11] Victor Chernozhukov, Iván Fernández-Val, and Blaise Melly. Inference on counterfactual distributions. *Econometrica*, 81(6):2205–2268, 2013.

[12] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[13] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[14] John DiNardo, Nicole M. Fortin, and Thomas Lemieux. Labor market institutions and the distribution of wages, 1973-1992: A semiparametric approach. *Econometrica*, 64(5):1001–1044, 1996.

[15] Garrett Fitzmaurice, Marie Davidian, Geert Verbeke, and Geert Molenberghs. *Longitudinal data analysis*. CRC press, 2008.

[16] Centers for Disease Control. Us state and territorial public mask mandates from april 10, 2020 through august 15, 2021 by county by day. *Policy Surveillance. September*, 10, 2021.

[17] Dennis Frauen, Tobias Hatt, Valentyn Melnychuk, and Stefan Feuerriegel. Estimating average causal effects from patient trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, number 6, pages 7586–7594, 2023.

[18] Keisuke Fujii, Koh Takeuchi, Atsushi Kuribayashi, Naoya Takeishi, Yoshinobu Kawahara, and Kazuya Takeda. Estimating counterfactual treatment outcomes over time in complex multi-agent scenarios. *arXiv preprint arXiv:2206.01900*, 2022.

[19] Google. Community mobility reports. `https://www.google.com/covid19/mobility/`, 2022. Accessed: 2022-09-15.

[20] Olivier Goudet, Diviyan Kalainathan, Philippe Caillou, Isabelle Guyon, David Lopez-Paz, and Michèle Sebag. Causal generative neural networks. *arXiv preprint arXiv:1711.08936*, 2017.

[21] CMU DELPHI Group. Covid-19 symptom surveys through facebook. `https://delphi.cmu.edu/blog/2020/08/26/covid-19-symptom-surveys-through-facebook/`, 2022. Accessed: 2022-09-15.

[22] Gery P Guy Jr, Florence C Lee, Gregory Sunshine, Russell McCord, Mara Howard-Williams, Lyudmyla Kompaniyets, Christopher Dunphy, Maxim Gakh, Regen Weber, Erin Sauber-Schatz, et al. Association of state-issued mask mandates and allowing on-premises restaurant dining with county-level covid-19 case and death growth rates—united states, march 1–december 31, 2020. *Morbidity and Mortality Weekly Report*, 70(10):350, 2021.

[23] Keisuke Hirano, Guido W Imbens, and Geert Ridder. Efficient estimation of average treatment effects using the estimated propensity score. *Econometrica*, 71(4):1161–1189, 2003.

[24] Daniel Jiwoong Im, Kyunghyun Cho, and Narges Razavian. Causal effect variational autoencoder with uniform treatment. *arXiv preprint arXiv:2111.08656*, 2021.

[25] Kosuke Imai and David A Van Dyk. Causal inference with general treatment regimes: Generalizing the propensity score. *Journal of the American Statistical Association*, 99(467):854–866, 2004.

[26] Guido W Imbens. Nonparametric estimation of average treatment effects under exogeneity: A review. *Review of Economics and statistics*, 86(1):4–29, 2004.

[27] Andrew Jesson, Sören Mindermann, Yarin Gal, and Uri Shalit. Quantifying ignorance in individual-level causal-effect estimates under hidden confounding. In *International Conference on Machine Learning*, pages 4829–4838. PMLR, 2021.

[28] E H Kennedy, S Balakrishnan, and L A Wasserman. Semiparametric counterfactual density estimation. *Biometrika*, page asad017, 03 2023.

[29] Kwangho Kim, Jisu Kim, and Edward H Kennedy. Causal effects based on distributional distances. *arXiv preprint arXiv:1806.02935*, 2018.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] Samantha Kleinberg and George Hripcsak. A review of causal inference for biomedical informatics. *Journal of biomedical informatics*, 44(6):1102–1112, 2011.

[32] Milan Kuzmanovic, Tobias Hatt, and Stefan Feuerriegel. Deconfounding temporal autoencoder: estimating treatment effects over time using noisy proxies. In *Machine Learning for Health*, pages 143–155. PMLR, 2021.

[33] Rui Li, Stephanie Hu, Mingyu Lu, Yuria Utsumi, Prithwish Chakraborty, Daby M Sow, Piyush Madan, Jun Li, Mohamed Ghalwash, Zach Shahn, et al. G-net: a recurrent network approach to g-computation for counterfactual prediction under a dynamic treatment regime. In *Machine Learning for Health*, pages 282–299. PMLR, 2021.

[34] Bryan Lim, Ahmed Alaa, and Mihaela van der Schaar. Forecasting treatment responses over time using recurrent marginal structural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[35] Jaechang Lim, Seongok Ryu, Jin Woo Kim, and Woo Youn Kim. Molecular generative model based on conditional variational autoencoder for de novo molecular design. *Journal of cheminformatics*, 10(1):1–9, 2018.

[36] Qiao Liu, Zhongren Chen, and Wing Hung Wong. Causalegm: a general causal inference framework by encoding generative modeling. *arXiv preprint arXiv:2212.05925*, 2022.

[37] 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.

[38] Helena C Maltezou, Androula Pavli, and Athanasios Tsakris. Post-covid syndrome: an insight on its pathogenesis. *Vaccines*, 9(5):497, 2021.

[39] Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes. In *International Conference on Machine Learning*, pages 15293–15329. PMLR, 2022.

[40] Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Normalizing flows for interventional density estimation. In *International Conference on Machine Learning*, pages 24361–24397. PMLR, 2023.

[41] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. cite arxiv:1411.1784.

[42] Ashish Mishra, Shiva Krishna Reddy, Anurag Mittal, and Hema A Murthy. A generative model for zero shot learning using conditional variational autoencoders. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2188–2196, 2018.

[43] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[44] Brady Neal, Chin-Wei Huang, and Sunand Raghupathi. Realcause: Realistic causal inference benchmarking. *arXiv preprint arXiv:2011.15007*, 2020.

[45] Jersey Neyman. Sur les applications de la théorie des probabilités aux experiences agricoles: Essai des principes. *Roczniki Nauk Rolniczych*, 10(1):1–51, 1923.

[46] My Nguyen. Mask mandates and covid-19 related symptoms in the us. *ClinicoEconomics and Outcomes Research*, pages 757–766, 2021.

[47] Artidoro Pagnoni, Kevin Liu, and Shangyan Li. Conditional variational autoencoder for neural machine translation. *arXiv preprint arXiv:1812.04405*, 2018.

[48] Harsh Parikh, Carlos Varjao, Louise Xu, and Eric Tchetgen Tchetgen. Validating causal inference methods. In *International Conference on Machine Learning*, pages 17346–17358. PMLR, 2022.

[49] Judea Pearl. Causal inference in statistics: An overview. 2009.

[50] Hadrien Reynaud, Athanasios Vlontzos, Mischa Dombrowski, Ciarán Gilligan Lee, Arian Beqiri, Paul Leeson, and Bernhard Kainz. D'artagnan: Counterfactual video generation. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2022: 25th International Conference, Singapore, September 18–22, 2022, Proceedings, Part VIII*, pages 599–609. Springer, 2022.

[51] James Robins. A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling*, 7(9-12):1393–1512, 1986.

[52] James Robins and Miguel Hernan. Estimation of the causal effects of time-varying exposures. *Chapman & Hall/CRC Handbooks of Modern Statistical Methods*, pages 553–599, 2008.

[53] James M Robins. Correcting for non-compliance in randomized trials using structural nested mean models. *Communications in Statistics-Theory and methods*, 23(8):2379–2412, 1994.

[54] James M Robins. Association, causation, and marginal structural models. *Synthese*, 121(1/2):151–179, 1999.

[55] James M Robins, Sander Greenland, and Fu-Chang Hu. Estimation of the causal effect of a time-varying exposure on the marginal mean of a repeated binary outcome. *Journal of the American Statistical Association*, 94(447):687–700, 1999.

[56] James M Robins, Miguel Angel Hernan, and Babette Brumback. Marginal structural models and causal inference in epidemiology. *Epidemiology*, pages 550–560, 2000.

[57] James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427):846–866, 1994.

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

[59] Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The annals of mathematical statistics*, pages 832–837, 1956.

[60] Donald B Rubin. Bayesian inference for causal effects: The role of randomization. *The Annals of statistics*, pages 34–58, 1978.

[61] Shiv Kumar Saini, Sunny Dhamnani, Akil Arif Ibrahim, and Prithviraj Chavan. Multiple treatment effect estimation using deep generative model with task embedding. In *The World Wide Web Conference*, pages 1601–1611, 2019.

[62] Axel Sauer and Andreas Geiger. Counterfactual generative networks. *arXiv preprint arXiv:2101.06046*, 2021.

[63] Peter Schulam and Suchi Saria. Reliable decision support using counterfactual models. *Advances in neural information processing systems*, 30, 2017.

[64] David W Scott and James R Thompson. Probability density estimation in higher dimensions. In *Computer Science and Statistics: Proceedings of the fifteenth symposium on the interface*, volume 528, pages 173–179. North-Holland, Amsterdam, 1983.

[65] Nabeel Seedat, Fergus Imrie, Alexis Bellot, Zhaozhi Qian, and Mihaela van der Schaar. Continuous-time modeling of counterfactual outcomes using neural controlled differential equations. *arXiv preprint arXiv:2206.08311*, 2022.

[66] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

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

[68] The New York Times. Coronavirus (covid-19) data in the united states. `https://github.com/nytimes/covid-19-data`, 2021. Accessed: 2022-09-15.

[69] Miriam E Van Dyke, Tia M Rogers, Eric Pevzner, Catherine L Satterwhite, Hina B Shah, Wyatt J Beckman, Farah Ahmed, D Charles Hunt, and John Rule. Trends in county-level covid-19 incidence in counties with and without a mask mandate—kansas, june 1–august 23, 2020. *Morbidity and Mortality Weekly Report*, 69(47):1777, 2020.

[70] Arnaud Van Looveren, Janis Klaise, Giovanni Vacanti, and Oliver Cobb. Conditional generative models for counterfactual explanations. *arXiv preprint arXiv:2101.10123*, 2021.

[71] Toon Vanderschueren, Alicia Curth, Wouter Verbeke, and Mihaela van der Schaar. Accounting for informative sampling when learning to forecast treatment outcomes over time. *arXiv preprint arXiv:2306.04255*, 2023.

[72] Lan Wang, Yu Zhou, Rui Song, and Ben Sherwood. Quantile-optimal treatment regimes. *Journal of the American Statistical Association*, 113(523):1243–1254, 2018.

[73] Yuxin Wang, Zicheng Deng, and Donglu Shi. How effective is a mask in preventing covid-19 infection? *Medical devices & sensors*, 4(1):e10163, 2021.

[74] Yongling Xiao, Michal Abrahamowicz, and Erica EM Moodie. Accuracy of conventional and marginal structural cox model estimators: a simulation study. *The international journal of biostatistics*, 6(2), 2010.

[75] 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.

[76] Weijia Zhang, Thuc Duy Le, Lin Liu, Zhi-Hua Zhou, and Jiuyong Li. Mining heterogeneous causal effects for personalized cancer treatment. *Bioinformatics*, 33(15):2372–2378, 2017.

[77] YiFan Zhang, Hanlin Zhang, Zachary Chase Lipton, Li Erran Li, and Eric Xing. Exploring transformer backbones for heterogeneous treatment effect estimation. In *NeurIPS ML Safety Workshop*, 2022.

[78] Shixiang Zhu, Alexander Bukharin, Liyan Xie, Mauricio Santillana, Shihao Yang, and Yao Xie. High-resolution spatio-temporal model for county-level covid-19 activity in the U.S. *ACM Trans. Manage. Inf. Syst.*, 12(4), sep 2021.

[79] Shixiang Zhu, Alexander Bukharin, Liyan Xie, Khurram Yamin, Shihao Yang, Pinar Keskinocak, and Yao Xie. Early detection of COVID-19 hotspots using spatio-temporal data. *IEEE Journal of Selected Topics in Signal Processing*, 16(2):250–260, 2022.

A Related work

Our work has connections to causal inference in time series, counterfactual density estimation, and generative models. To our best knowledge, our work is the first to intersect the three aforementioned areas. Below we review each of these areas independently.

Causal inference with time-varying treatments. Causal inference has historically been related to longitudinal data. Classic approaches to analyzing time-varying treatment effects include the g-computation formula, structural nested models, and marginal structural models [60, 51, 53, 57, 56, 15, 33]. These seminal works are typically based on parametric models with limited flexibility. Recent advancements in machine learning have significantly accelerated progress in this area using flexible statistical models [63, 10] and deep neural networks [34, 6, 4, 33, 65, 39, 17, 71] to capture the complex temporal dependency of the outcome on treatment and covariate history. These approaches, however, focus on predicting the mean counterfactual outcome instead of the distribution. The performance of these methods also heavily relies on the specific structures (*e.g.*, LSTMs) without more flexible architectures.

Counterfactual distribution estimation. Recently, several approaches have emerged to estimate the entire counterfactual distribution rather than the means, including estimating quantiles of the cumulative distributional functions (CDFs) [11, 72], re-weighted kernel estimations [14], and semiparametric methods [28]. In particular, [28] highlights the extra information afforded by estimating the entire counterfactual distribution and using the distance between counterfactual densities as a measure of causal effects. [40] uses normalizing flow to estimate the interventional density. However, these methods are designed to work under static settings with no time-varying treatments [2], and are explicit density estimation methods that may be difficult to scale to high-dimensional outcomes. [33] proposes a deep framework based on G-computation which can be used to simulate outcome trajectories on which one can estimate the counterfactual distribution. However, this framework approximates the distribution via empirical estimation of the sample variance, which may be unable to capture the complex variability of the (potentially high-dimensional) distributions. Our work, on the other hand, approximates the counterfactual distribution with a generative model without explicitly estimating its density. This will enable a wider range of application scenarios including continuous treatments and can accommodate more intricate data structures in the high-dimensional outcome settings.

Counterfactual generative model. Generative models, including a variety of deep network architectures such as generative adversarial networks (GAN) and autoencoders, have been recently developed to perform counterfactual prediction [20, 37, 75, 61, 44, 62, 70, 24, 32, 3, 18, 36, 50, 77, 48]. However, many of these approaches primarily focus on using representation learning to improve treatment effect estimation rather than obtaining counterfactual samples or approximating counterfactual distributions. For example, [75, 61] adopt deep generative models to improve the estimation of individual treatment effects (ITEs) under static settings. Some of these approaches focus on exploring causal relationships between components of an image [62, 70, 50]. Furthermore, there has been limited exploration of applying generative models to time series settings in the existing literature. A few attempts, including [37, 32], train autoencoders to estimate treatment effect using longitudinal data. Nevertheless, these methods are not intended for drawing counterfactual samples. In sum, to the best of our knowledge, our work is the first to use generative models to approximate counterfactual distribution from data with time-varying treatments, a novel setting not addressed by prior works.

B Assumptions

The standard assumptions needed for identifying the treatment effects are [15, 34, 63]:

1. *Consistency*: If $\overline{A}_t = \overline{a}_t$ for a given subject, then the counterfactual outcome for treatment, $\overline{a}_t$, is the same as the observed (factual) outcome: $Y(\overline{a}_t) = Y$.
2. *Positivity*: If $\mathbb{P}\{\overline{A}_{t-1} = \overline{a}_{t-1}, \overline{X}_t = \overline{x}_t\} \neq 0$, then $\mathbb{P}\{\overline{A}_t = \overline{a}_t | \overline{A}_{t-1} = \overline{a}_{t-1}, \overline{X}_t = \overline{x}_t\} > 0$ for all $\overline{a}_t$ [25].
3. *Sequential strong ignorability*: $Y(\overline{a}_t) \perp\!\!\!\perp \overline{A}_t | \overline{A}_{t-1} = \overline{a}_{t-1}, \overline{X}_t = \overline{x}_t$, for all a_t and t.

Assumption 2 means that, for each timestep, each treatment has a non-zero probability of being assigned. Assumption 3 (also called conditional exchangeability) means that there are no unmeasured

confounders, that is, all of the covariates affecting both the treatment assignment and the outcomes are present in the the observational dataset. Note that while assumption 3 is standard across all methods for estimating treatment effects, it is not testable in practice [49, 56].

C Proof of Lemma 1

Given a probability distribution for $(Y, \overline{A}, \overline{X})$ and a causal directed acyclic graph (DAG) shown in Figure 7, we can factor $f(Y, \overline{A}, \overline{X})$ as

$$f(Y, \overline{A}, \overline{X}) = f\left(Y|\overline{A}, \overline{X}\right) \prod_{\tau=t-d}^{t} f\left(X_\tau|\overline{A}_{\tau-1}, \overline{X}_{\tau-1}\right) \prod_{\tau=t-d}^{t} f\left(A_\tau|\overline{A}_{\tau-1}, \overline{X}_\tau\right). \tag{7}$$

Using the definition of g-formula [54], we have

$$\begin{aligned}
f_{\overline{a}}(y) &= \int f\left(y|\overline{a}, \overline{X}\right) \cdot \prod_{\tau=t-d}^{t} f\left(X_\tau|\overline{a}_{\tau-1}, \overline{X}_{\tau-1}\right) d\overline{X} \\
&= \int f\left(y|\overline{a}, \overline{X}\right) \cdot \frac{\prod_{\tau=t-d}^{t} f\left(a_\tau|\overline{a}_{\tau-1}, \overline{X}_\tau\right)}{\prod_{\tau=t-d}^{t} f\left(a_\tau|\overline{a}_{\tau-1}, \overline{X}_\tau\right)} \cdot \prod_{\tau=t-d}^{t} f\left(X_\tau|\overline{a}_{\tau-1}, \overline{X}_{\tau-1}\right) d\overline{X} \\
&\overset{(i)}{=} \int \frac{1}{\prod_{\tau=t-d}^{t} f\left(a_\tau|\overline{a}_{\tau-1}, \overline{X}_\tau\right)} f\left(y, \overline{a}, \overline{X}\right) d\overline{X} \\
&= \int \frac{\mathbb{1}\{\overline{A} = \overline{a}\}}{\prod_{\tau=t-d}^{t} f\left(A_\tau|\overline{A}_{\tau-1}, \overline{X}_\tau\right)} f\left(y, \overline{A}, \overline{X}\right) d\overline{A} d\overline{X},
\end{aligned}$$

where the equation (i) holds due to (7).

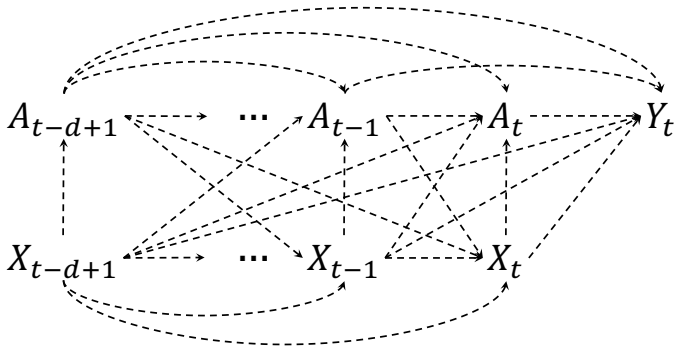


Figure 7: The causal directed acyclic graph (DAG) of the time-varying treatment.

D Proof of Proposition 1

Note that the unstabilized weight is defined as $w(\overline{A}, \overline{X}) = 1/\prod_{\tau=t-d}^{t} f\left(A_\tau | \overline{A}_{\tau-1}, \overline{X}_\tau\right)$. Using Lemma 1, we have

$$
\begin{aligned}
\mathop{\mathbb{E}}_{y\sim f_{\overline{a}}} \log f_\theta(y|\overline{a}) &= \int \log f_\theta(y|\overline{a}) f_{\overline{a}}(y)\, dy \\
&= \int \log f_\theta(y|\overline{a}) \int \frac{\mathbb{1}\{\overline{A}=\overline{a}\}}{\prod_{\tau=t-d}^{t} f\left(A_\tau|\overline{A}_{\tau-1}, \overline{X}_\tau\right)} f\left(y, \overline{A}, \overline{X}\right) d\overline{A} d\overline{X} dy \\
&= \int \log f_\theta(y|\overline{a}) \int w(\overline{A}, \overline{X}) \mathbb{1}\{\overline{A}=\overline{a}\} f\left(y, \overline{A}, \overline{X}\right) d\overline{A} d\overline{X} dy \\
&= \int \log f_\theta(y|\overline{a}) \int w(\overline{a}, \overline{X}) f\left(y, \overline{a}, \overline{X}\right) d\overline{X} dy \\
&= \int\int \log f_\theta(y|\overline{a}) w(\overline{a}, \overline{X}) f\left(y, \overline{a}, \overline{X}\right) d\overline{X} dy \\
&= \int\int \log f_\theta(y|\overline{a}) w(\overline{a}, \overline{X}) f\left(y, \overline{a}|\overline{X}\right) f\left(\overline{X}\right) dy d\overline{X} \\
&\overset{(i)}{=} \int w(\overline{a}, \overline{X}) f\left(\overline{X}\right) \mathop{\mathbb{E}}_{(y,\overline{a})} [\log f_\theta(y|\overline{a}) \big| \overline{X}] d\overline{X} \\
&= \mathop{\mathbb{E}}_{\overline{X}} \left[\mathop{\mathbb{E}}_{(y,\overline{a})} \left[w(\overline{a}, \overline{X}) \log f_\theta(y|\overline{a}) \big| \overline{X} \right] \right] \\
&\overset{(ii)}{=} \mathop{\mathbb{E}}_{(y,\overline{a},\overline{x})\sim f} w(\overline{a}, \overline{x}) \log f_\theta(y|\overline{a}) \\
&\approx \sum_{(y,\overline{a},\overline{x})\in\mathcal{D}} w(\overline{a}, \overline{x}) \log f_\theta(y|\overline{a}),
\end{aligned}
$$

where (i) follows from Fubini's theorem and (ii) follows from the tower property of expectation.

E Connection to counterfactual density estimation

Plug-in density estimation Plug-in approaches have been commonly used to estimate the counterfactual density in the static setting[7, 29, 28] and can be extended to our time-varying setting via direct application of Lemma 1. However, this practice could be problematic when the sample size is large as it requires averaging the entire observed dataset for each evaluation of y. Instead, we circumvent this computational challenge by approximating the counterfactual density using a proxy conditional distribution $f_\theta(\cdot|\overline{a})$ which is represented by a generative model, $g_\theta(z, \overline{a})$.

Doubly-robust (semi) parametric density estimators Doubly-robust density estimators have proven successful in directly estimating the counterfactual density in the static setting [28, 40]. Our framework differs from these methods in three aspects:

1. To our best knowledge, there is a scarcity of unified theory for doubly-robust density approximation of potential outcomes in longitudinal settings. One may wish to extend our framework to a doubly robust setting, and a common approach is to incorporate an estimator including G-computation [52, 33] into the loss function. When Y is potentially high-dimensional, however, correct estimation of the outcome model and the covariate density model in G-computation become challenging. Therefore, we opt for the IPTW-based approach in proposition 1 as estimating the propensity model is less challenging thanks to the 1-dimensional, binary values of A_t.
2. The direct density estimation approaches in [28, 40] use a separate density model to directly approximate $f_\theta(\cdot|\overline{a})$ for each $\overline{a}$, whereas our approach uses a generator, $g_\theta(z, \overline{a})$ to approximate the proxy conditional distribution $f_\theta(\cdot|\overline{a})$ under all $\overline{a}$. This approach requires training only a single model and has the potential to generalize to continuous treatments.
3. The framework in [40], when extended to the time-varying scenario using IPTW, requires integrating the log-likelihood of the density model over both the observed samples and the

outcome space $\mathcal{Y}$ (see (8)). In practice, this will require performing a Monte Carlo sampling of Y for each gradient step to optimize (8), which can be prohibitive when $\mathcal{Y}$ is high-dimensional. Our proposed loss function in Proposition 1, on the other hand, only requires computing the weighted log-likelihood over observed samples which is easy to implement. Therefore, our Proposition 1 can be viewed as a novel reformulation of (8) that enhances the scalability of model training for high-dimensional outcomes.

$$\mathop{\mathbb{E}}_{y\sim f_{\overline{a}}}[-\log f_\theta(y)] \approx \int_{y\in\mathcal{Y}} \log f_\theta(y) \sum_{(y,\overline{a},\overline{x})\in\mathcal{D}} w_\phi(\overline{a},\overline{x}) f(y,\overline{a},\overline{x}) dY. \tag{8}$$

F Derivation and implementation details of variational learning

Derivation of the proxy conditional distribution Now we present the derivation of the log conditional probability density function (PDF) in (6). To begin with, it can be written as:

$$\log f_\theta(y|\overline{a}) = \log \int p_\theta(y,z|\overline{a}) dz,$$

where z is a latent random variable. This integral has no closed form and can be usually estimated by Monte Carlo integration with importance sampling, *i.e.*,

$$\int p_\theta(y,z|\overline{a}) dz = \mathop{\mathbb{E}}_{z\sim q(\cdot|y,\overline{a})}\left[\frac{p_\theta(y,z|\overline{a})}{q(z|y,\overline{a})}\right].$$

Here $q(z|y,\overline{a})$ is the proposed variational distribution, where we can draw sample z from this distribution given y and $\overline{a}$. Therefore, by Jensen's inequality, we can find the evidence lower bound (ELBO) of the conditional PDF:

$$\log f_\theta(y|\overline{a}) = \log \mathop{\mathbb{E}}_{z\sim q(\cdot|y,\overline{a})}\left[\frac{p_\theta(y,z|\overline{a})}{q(z|y,\overline{a})}\right] \geq \mathop{\mathbb{E}}_{z\sim q(\cdot|y,\overline{a})}\left[\log\frac{p_\theta(y,z|\overline{a})}{q(z|y,\overline{a})}\right].$$

Using Bayes rule, the ELBO can be equivalently expressed as:

$$\begin{aligned}
\mathop{\mathbb{E}}_{z\sim q(\cdot|y,\overline{a})}\left[\log\frac{p_\theta(y,z|\overline{a})}{q(z|y,\overline{a})}\right] &= \mathop{\mathbb{E}}_{z\sim q(\cdot|y,\overline{a})}\left[\log\frac{p_\theta(y|z,\overline{a})p_\theta(z|\overline{a})}{q(z|y,\overline{a})}\right] \\
&= \mathop{\mathbb{E}}_{z\sim q(\cdot|y,\overline{a})}\left[\log\frac{p_\theta(z|\overline{a})}{q(z|y,\overline{a})}\right] + \mathop{\mathbb{E}}_{z\sim q(\cdot|y,\overline{a})}\left[\log p_\theta(y|z,\overline{a})\right] \\
&= -D_{\mathrm{KL}}(q(z|y,\overline{a})||p_\theta(z|\overline{a})) + \mathop{\mathbb{E}}_{z\sim q(\cdot|y,\overline{a})}\left[\log p_\theta(y|z,\overline{a})\right].
\end{aligned}$$

Implementation details For the KL-divergence term in the ELBO (6), both $q(z|y,\overline{a})$ and $p_\theta(z|\overline{a})$ are often modeled as Gaussian distributions, which allows us to compute the KL divergence of Gaussians with a closed-form expression. In practice, we introduce two additional generators, including the encoder net $g_{\text{encode}}(\epsilon,y,\overline{a})$ and the prior net $g_{\text{prior}}(\epsilon,\overline{a})$, respectively, to represent $q(z|y,\overline{a})$ and $p_\theta(z|\overline{a})$ as transformations of another random variable $\epsilon\sim\mathcal{N}(0,I)$ using reparameterization trick [66]. A common choice is a simple factorized Gaussian encoder. For example, the approximate posterior $q(z|y,\overline{a})$ can be represented as:

$$q(z|y,\overline{a}) = \mathcal{N}(z;\mu,\mathrm{diag}(\Sigma)),$$

or

$$q(z|y,\overline{a}) = \prod_{j=1}^{r} q(z_j|y,\overline{a}) = \prod_{j=1}^{r}\mathcal{N}(z_j;\mu_j,\sigma_j^2).$$

The Gaussian parameters $\mu = (\mu_j)_{j=1,\dots,r}$ and $\mathrm{diag}(\Sigma) = (\sigma_j^2)_{j=1,\dots,r}$ can be obtained using reparameterization trick via an encoder network ϕ:

$$\begin{aligned}
(\mu, \log\mathrm{diag}(\Sigma)) &= \phi(y,\overline{a}), \\
z &= \mu + \sigma\odot\epsilon,
\end{aligned}$$

Algorithm 1 Learning algorithm for the conditional generator θ

Input: Training set $\mathcal{D}$ data tuples: $\mathcal{D} = \{(y_t^{(i)}, \overline{a}_t^{(i)}, \overline{x}_t^{(i)})\}_{t=d,\ldots,T,\ i=1,\ldots,N}$ where T is the time horizon and I is the total number of individuals; the number of the learning epoches E.
Initialization: model parameters θ and fitted $\widehat{\phi}$ using $\mathcal{D}$.
while $e < E$ **do**
  for each sampled batch $\mathcal{D}^k$ with size n **do**
    1. Draw samples $\epsilon \sim \mathcal{N}(0, I)$ from noise distribution;
    2. Compute the ELBO of $\log f_\theta(y|\overline{a})$ for $(y, \overline{a}, \overline{x}) \in \mathcal{D}^k$ given ϵ and θ according to (6);
    3. Re-weight the ELBO for $(y, \overline{a}, \overline{x}) \in \mathcal{D}^k$ using $w_{\widehat{\phi}}(\overline{a}, \overline{x})$ according to (5);
    4. Update θ using stochastic gradient descent by maximizing (4).
  end for
end while
return θ

where $\epsilon \sim \mathcal{N}(0, I)$ is another random variable and $\odot$ is the element-wise product. Because both $q(z|y_i, \overline{a}_{i-1})$ and $p_\theta(z|\overline{a}_{i-1})$ are modeled as Gaussian distributions, the KL divergence can be computed using a closed-form expression.

The log-likelihood of the second term can be implemented as the reconstruction loss and calculated using generated samples. Maximizing the negative log-likelihood $p_\theta(y|z, \overline{a})$ is equivalent to minimizing the cross entropy between the distribution of an observed outcome y and the reconstructed outcome $\widetilde{y}$ generated by the generative model g given z and the history $\overline{a}$.

We emphasize that our model is not tied to any specific type of generative models and learning algorithms, and we use the variational learning framework for illustrative purposes.

G Additional experiment details

G.1 Baselines

Here we present an additional review of each baseline method in the paper as well as implementation details.

Marginal structural model with a fully-connected neural network (`MSM+NN`) We include the classic `MSM+NN` proposed in [57, 51]. This classical framework assumes that the counterfactual mean of the outcome variable can be represented as a linear function of the treatments. We use this model while replacing the linear model with a 3-layer fully-connected neural network, g_{msm}. This serves as a deterministic baseline for our generative framework. We learn the `MSM+NN` using stochastic gradient descent with a weighted loss function:

$$\sum_{(y,\overline{a},\overline{x})\in\mathcal{D}} w_\phi(\overline{a}, \overline{x})(y - g_{\text{msm}}(\overline{a}))^2.$$

To establish a fair comparison, we train the `MSM+NN` using an identical training size to that of the `MSCVAE` model. We train the `MSM+NN` for $1,000$ epochs with a learning rate of 0.01. However, it's important to note that in this particular setup, our capacity is limited to estimating the mean instead of the entire distribution. For computing the Wasserstein distance in the full-synthetic experiments, we treat the `MSCVAE` samples as coming from a degenerate distribution at its predicted value.

Conditional variational autoencoder (`CVAE`) To examine the impact of Inverse Probability of Treatment Weighting (IPTW) on training generative models, we include a vanilla conditional variational autoencoder (`CVAE`) with an architecture identical to that of the `MSCVAE`, but excluding IPTW weighting. The `CVAE` is a widely-used type of conditional generative model that has found applications in various tasks, including image generation [42, 67], neural machine translation [47], and molecular design [35]. To train the `CVAE`, we follow the same procedure as `MSCVAE`, with the exception that we replace the loss function with the unweighted version of (4).

$$\sum_{(y,\overline{a},\overline{x})\in\mathcal{D}} \log f_\theta(y|\overline{a}),$$

where $f_\theta(\cdot)$ is the conditional distribution represented by the `CVAE`.

Kernel density estimator (`KDE`) We use a Gaussian kernel density estimator [59] to estimate the empirical conditional distribution from the observed data. This is achieved by running KDE on the observed outcomes with the same treatments, *i.e.*,

$$f_{\overline{a}} \approx g_{\text{kde}}(y|\overline{A} = \overline{a}),$$

where $g_{\text{kde}}(\cdot)$ is the KDE estimator. We learn the KDE with bandwidth set to 0.5, 1, 1.5, and 2, respectively, and report the metrics with bandwidth $= 0.5$ as the optimal results.

Semi-parametric Plug-in method based on pseudo-population (`Plugin+KDE`) We include a baseline using Lemma 1 as a plugin estimator by following the semi-parametric KDE approach in [40]. Specifically, we rewrite Lemma 1 as:

$$f_{\overline{a}}(y) \approx \sum_{(y,\overline{a},\overline{x})\in\mathcal{D}} \mathbb{1}\{\overline{A} = \overline{a}\} w_\phi(\overline{a},\overline{x}) f\left(y, \overline{A}, \overline{x}\right).$$

To estimate the right-hand side of the equation, we performed KDE on $y|\overline{A} = \overline{a}$ where each sample tuple $(y, \overline{a}, \overline{x})$ is weighted by its IPTW, $w_\phi(\overline{a}, \overline{x})$, for each $\overline{A} = \overline{a}$ separately. The bandwidth is set to be the same as in `KDE`.

G-Net (`G-Net`) We implement `G-net` proposed in [33] based on G-computation. For our experiment setting, at each time step $t \in [T]$, we designed the conditional covariates block, the history representation block, and the final conditional outcome block as a 3-layer fully connected neural network respectively. The types of blocks are interconnected to form sequential net structures across different time steps, followed by a conditional outcome block at the end, which has a 2-layer structure. This makes the `G-net` model include a total of $(2 \times d) + 1$ blocks. The loss function is the sum of the mean squared error:

$$\sum_{(\overline{x},y)\in\mathcal{D}} (\widehat{\overline{x}} - \overline{x})^2 + (\widehat{y} - y)^2,$$

where $\widehat{\overline{x}}$ and $\overline{x}$ are the predicted and groundtruth covariate history, while $\widehat{y}$ and y are the predicted and groundtruth outcome. Following the original literature, we impose a Gaussian parametric assumption over the underlying counterfactual distribution, and introduce prediction variability by adding Gaussian noise whose variance is empirically estimated from the residuals between the predicted and groundtruth outcomes.

G.2 Experiment metrics

To quantify the quality of the approximated counterfactual distributions, we used the following metrics:

Mean This is the difference between the empirical mean of the evaluated samples.

1-Wasserstein Distance We used the earth mover's distance, which is defined as:

$$l_1(u, v) = \inf_{\pi\in\Gamma(u,v)} \int_{\Omega\times\Omega} |x - y| d\pi(x, y),$$

where $\Gamma(u, v)$ is the joint probability distributions for the groundtruth and learned counterfactual distributions, and Ω is the space of each distribution.

FID* Both semi-synthetic datasets have high-dimensional outcomes, making comparisons using the mean or Wasserstein distance of the distributions less interpretable. A common approach in the generative model community is FID (Fréchet inception distance). In summary, FID uses a pre-trained neural network (frequently the inception v3 model) to obtain a feature vector for each sample, generated for groundtruth. The feature vector is the activation of the last pooling layer prior to the output layer of the pre-trained network. The feature vectors are then summarized as

multivariate Gaussians by computing their mean and covariances. The distance between the generated or groundtruth image distribution is then computed by calculating the 2-Wasserstein distance between two sets of Gaussians. A lower FID score represents a more realistic distribution for the generated images.

Since FID is not specifically designed for our TV-MNIST and semi-synthetic COVID-19 datasets, we propose to use FID* by following a similar idea of FID. For the semi-synthetic COVID-19 dataset, we first compute a PCA projection matrix of size 67×2 using samples from the counterfactual distribution under each treatment. The projection serves as the purpose of the pre-trained network in the original FID because it captures key information, including spatial correlation, of the 67-dimensional outcome variables. For each treatment combination, we then project the 67-dimensional samples into the 2-dimensional representational space using the PCA projection matrix and compute the 1-Wasserstein distance of the projection between the generated and counterfactual samples. A lower FID* score represents the generated samples have a similar distribution compared to the counterfactual ones.

For the TV-MNIST dataset, we use a 3-layer fully-connected neural network pre-trained to classify MNIST images. This network serves as the purpose of the pre-trained network in the original FID because it represents the semantic information (the digit label) of the 784-dimensional outcome variables. For each treatment combination, we then project the 784-dimensional samples into a 1-dimensional label space using the pre-trained MNIST classifier and compute the 1-Wasserstein distance of the projection between the generated and counterfactual samples. A lower FID* score represents the generated samples have a similar semantic distribution (in terms of the digit labels) compared to the counterfactual ones.

G.3 Experiment set-up

Experiment set-up To learn the model parameter θ, we use stochastic gradient descent to maximize the weighted log-likelihood (4). We adopt an Adam optimizer [30] with a learning rate of 10^{-3} and a batch size of 256. To ensure learning stability, we follow a commonly-used practice [74, 34] that involves truncating the subject-specific IPTW weights at the 0.01-th and 99.99-th percentiles and normalizing them by their mean. All experiments are performed on Jupyter Notebook with 16GB RAM and a 2.6 GHz 6-Core Intel Core i7 CPU.

The counterfactual generator g_θ, the IPTW w_ϕ, and the encoder network g_{encode} share the same two-layer fully-connected network architecture with ReLU activation. The layer width is set to $1,000$, and the length of the latent variable z is set to r which is determined by the specific synthetic experiment setting: $r = 5$ for $d = 1$ and $d = 3$, $r = 10$ for $d = 5$ and all the semi-synthetic and real data. For g_{encode}, the fully-connected networks map the $d + 1$ dimensional input vector (consisting of a d-dimensional treatment and 1-dimensional response) to the r-dimensional latent representation. For g_θ, the fully-connected networks map the $r + d$ dimensional input vector (consisting of a d-dimensional treatment and r-random noise) to the 1-dimensional generated counterfactual outcome. For w_ϕ, the fully-connected networks map the $2d$-dimensional input vector (consisting of a d-dimensional treatment and d-dimensional covariate) to the 1-dimensional conditional probability. We use a Sigmoid output layer for w_ϕ to ensure the output falls within $[0, 1]$. We set the batch size to 256 and the number of training epochs to 200 for training all the models in both synthetic and real data settings. The learning rate was set to 10^{-3} with a linear step-wise learning rate scheduler (Pytorch learning rate scheduler function `StepLR`) to ensure stable convergence of the learning process.

G.4 Fully Synthetic data

In this section, we provide an overview of the procedures for generating synthetic data. Our goal is to evaluate the performance of the proposed `MSCVAE` method and compare it to baseline approaches in the context of time-varying treatments. We follow the classic setting in [55] and simulate time series data with time-varying treatments and covariates. The presence of the time-varying confounders serves as an appropriate testbed for comparing MSM-based models to the baselines. To be specific, we generate three synthetic datasets with varying levels of historical dependence denoted as d. Each dataset consists of 10,000 trajectories, which represent recorded observations of individual subjects. These trajectories comprise 100 data tuples, encompassing treatment, covariate, and outcome values at specific time points. The causal relationships between these variables are visually depicted in Figure 7. For each time trajectory of length T, the datasets are generated based on the following

Table 2: Coefficients of the linear model in synthetic data generation

	α	β	γ
$d=1$	$(-3, 2, -1)$	$(-0.5, 0.5, 0.5, -0.5)$	$(0, 1, -1)$
$d=3$	$(-1, 12, 6, 3, 2, 1, 0.5)$	$(-0.5, 0.5, -0.5, 0.5, 0.5, -0.5, 0.5, -0.5)$	$(-1, 1.5, 1, 0.5, -1.5, -1, -0.5)$
$d=5$	$(-1, 12, 6, 3, 1, 0.5, 2, 1, 0.5, 0.1, 0.05)$	$(-0.5, 0.5, -0.5, 0.5, -0.5, 0.5, 0.5, -0.5, 0.5, -0.5, 0.5, -0.5)$	$(-1, 1.5, 1, 0.5, 0.1, 0.05, -1.5, -1, -0.5, -0.1, -0.05)$

Algorithm 2 Algorithm for obtaining a counterfactual sample

Input: Generated trajectory of a single subject: $\{(Y_t, X_t, A_t)\}_{t=1,\cdots,T}$.
Initialization: Given the treatment history $\overline{A}_T = \overline{a}$.
for $\tau = T - d + 1 : T$ **do**
1. Generate the covariate x_τ based on $\overline{A}_{\tau-1}$ and $\overline{X}_{\tau-1}$ according to (10).
2. Update the covariate $\overline{X}_\tau \leftarrow x_\tau$.

end for
Generate $Y(\overline{a})$ based on $\overline{A}_T$ and $\overline{X}_T$ according to (12).
return $Y(\overline{a})$

equations:

$$X_0 \sim \text{uniform}(0, 1), \tag{9}$$

$$X_t = \gamma_0 + \sum_{\tau=t-d}^{t-1} \gamma_{t-\tau} A_\tau + \sum_{\tau=t-d}^{t-1} \gamma_{d+t-\tau} X_\tau, \tag{10}$$

$$\mathbb{P}\{A_t = 1\} = \sigma(\beta_0 + \sum_{\tau=t-d}^{t-1} \beta_{t-\tau} A_\tau + \sum_{\tau=t-d}^{t} \beta_{d+t-\tau} X_\tau), \tag{11}$$

$$Y_t = \alpha_0 + \sum_{\tau=t-d}^{t-1} \alpha_{t-\tau} A_t + \sum_{\tau=t-d}^{t-1} \alpha_{d+t-\tau} X_\tau + \epsilon, \tag{12}$$

where $\epsilon \sim \mathcal{N}(0, 0.05)$ is the observation noise and $\sigma(\cdot)$ is a Sigmoid function. The specific coefficients are set according to the values in Table 2 to ensure the generation of valid synthetic data distributions with diversity:

Adjusting β_0 will change the balance of the treatment combinations: when keeping the remaining β coefficients, treatment variables $\overline{a}$, and covariates $\overline{x}$ unchanged, a smaller value of β_0 reduces the probability of treatment exposure, *i.e.*, $\mathbb{P}(A_t = 1)$. Consequently, this lower probability of treatment exposure results in a decrease in the occurrence of treatment combinations with exposures, leading to an imbalanced ratio among different treatment combinations. In Figure. 4, we set $\beta_0 = -0.5$ which results in an approximated balanced number of samples per treatment combination. In Appendix H, we include a figure by setting $\beta_0 = -2$, as a visualization of imbalanced treatment combinations.

To ensure the validity of our synthetic data generation process, we verify that the three assumptions outlined in Appendix B are satisfied. Assumptions 1 and 3 are naturally met because the ground truth model guarantees that the counterfactual outcome equals the observed outcome and that there are no unmeasured confounders. As for assumption 2, since the conditional probability of treatment is the Sigmoid function applied to a finite linear combination of historical treatments and covariates, it will always be positive.

Once the synthetic data is generated, we obtain counterfactual distributions to assess the performance of our proposed method. Specifically, we use the synthetic data to obtain samples from the counterfactual outcome distribution, $Y(\overline{a})$, for any given treatment combination $\overline{a}$. This is achieved by iteratively fixing the treatment sequence in the time series and generating the covariates and response variables according to equations (10) and (12) for each of the $10,000$ trajectories. The detailed procedure for obtaining a single counterfactual outcome sample is summarized in Algorithm 2.

G.5 Semi-synthetic time-varying MNIST data

We provide a benchmark based on the MNIST dataset. Specifically, the outcomes are MNIST images ($m = 784$). First, we compute a one-dimensional summary, the ϕ score [27], using each MNIST

Table 3: Real data description

Name	Description	Min	Max	Mean	Median	Std
Y	county-wise incremental new cases count ($\log_{10}$)	0	1.15×10^{-1}	2×10^{-3}	1×10^{-3}	2.7×10^{-3}
A	county-wise mask mandate	0	1×10^{0}	5.35×10^{-1}	1×10^{0}	4.99×10^{-1}
$X^{(0)}$	county-wise incremental death cases count ($\log_{10}$)	0	3.12×10^{-3}	3×10^{-4}	0	9×10^{-5}
$X^{(1)}$	county-wise average retail and recreation	-5.45×10^{1}	2.23×10^{1}	-4.27×10^{0}	-3.33×10^{0}	6.16×10^{0}
$X^{(2)}$	county-wise symptom value	0	3.23×10^{1}	9.3×10^{-1}	8.1×10^{-1}	5.1×10^{-1}

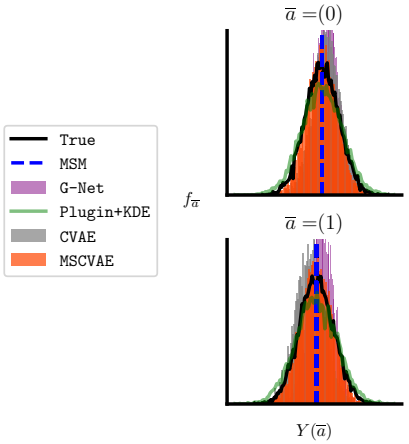


Figure 8: The estimated and true counterfactual distributions for $d = 1$ on synthetic datasets.

image. The ϕ value of an image depends on its average light intensity and its digit label. We refer the readers to [27] for the details on computing ϕ. Here we set the length of history dependence, d, to 3. We then define a linear model of 1-dimensional latent process to G.4 and simulate $1,000$ trajectories of the (X, A, Y) tuples of 100 time points according to the following equations:

$$X_0 \sim \text{uniform}(0, 1), \tag{13}$$

$$X_t = \gamma_0 + \sum_{\tau=t-2}^{t-1} \gamma_{t-\tau} A_\tau + \sum_{\tau=t-2}^{t-1} \gamma_{t-\tau+3} X_\tau, \tag{14}$$

$$\mathbb{P}\{A_t = 1\} = \sigma(\beta_0 + \sum_{\tau=t-2}^{t-1} \beta_{t-\tau} A_\tau + \sum_{\tau=t-2}^{t} \beta_{t-\tau+3} X_\tau), \tag{15}$$

$$\phi_t = 0.5 \left\lceil 10\sigma(\alpha_0 + \sum_{\tau=t-3}^{t-1} \alpha_{t-\tau} A_t + \sum_{\tau=t-3}^{t-1} \alpha_{t-\tau+3} X_\tau) - 0.6 \right\rceil, \tag{16}$$

$$Y_t \sim \{\texttt{MNIST}(i) : i = \arg\min |\phi_i - \phi_t|\}, \tag{17}$$

where $\sigma(\cdot)$ is a Sigmoid function, $\lceil \cdot \rceil$ is the ceiling function, and $\texttt{MNIST}(i)$ represents the MNIST image indexed by i. The coefficients are set according to Table 2 to ensure the generation of diverse data distributions. We generate the counterfactual samples according to Algorithm 2 by replacing the corresponding propensity and outcome models with the formulations above. The generated observations and counterfactual samples under the same treatment combinations may correspond to MNIST images of different labels. This way we can qualitatively assess the performance of an algorithm by comparing the labels of the MNIST images it generates against the counterfactual samples, as in as in Figure. 5a.

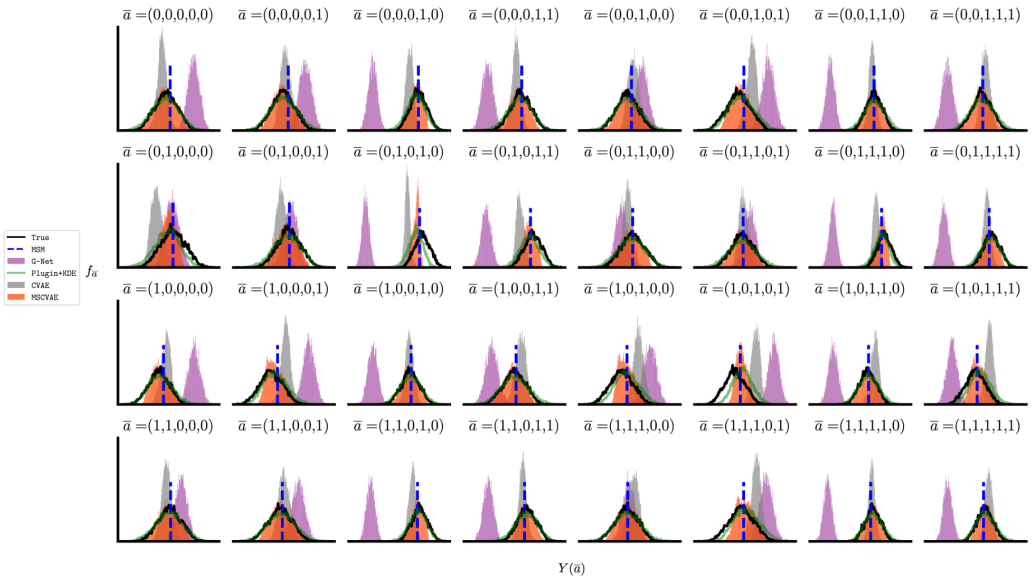


Figure 9: The estimated and true counterfactual distributions for $d = 5$ on synthetic datasets.

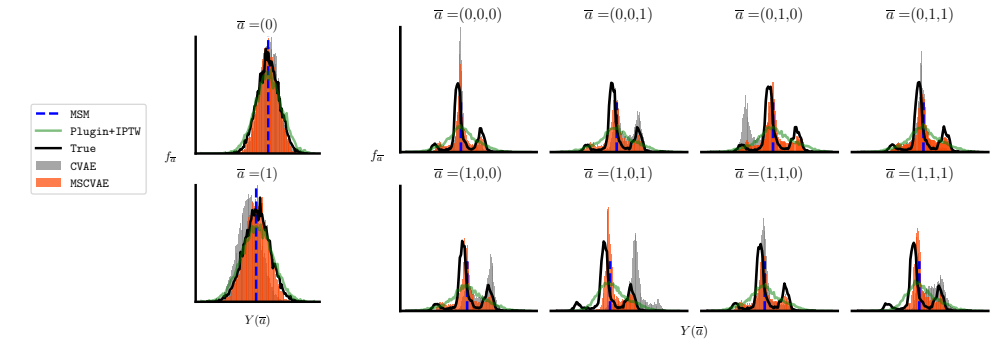
G.6 COVID-19 data

Since both the semi-synthetic Pennsylvania COVID-19 mask data and the real nationwide COVID-19 mask datasets are based on the same set of aggregated sources. We first introduce the data sources and then include the details of each dataset respectively.

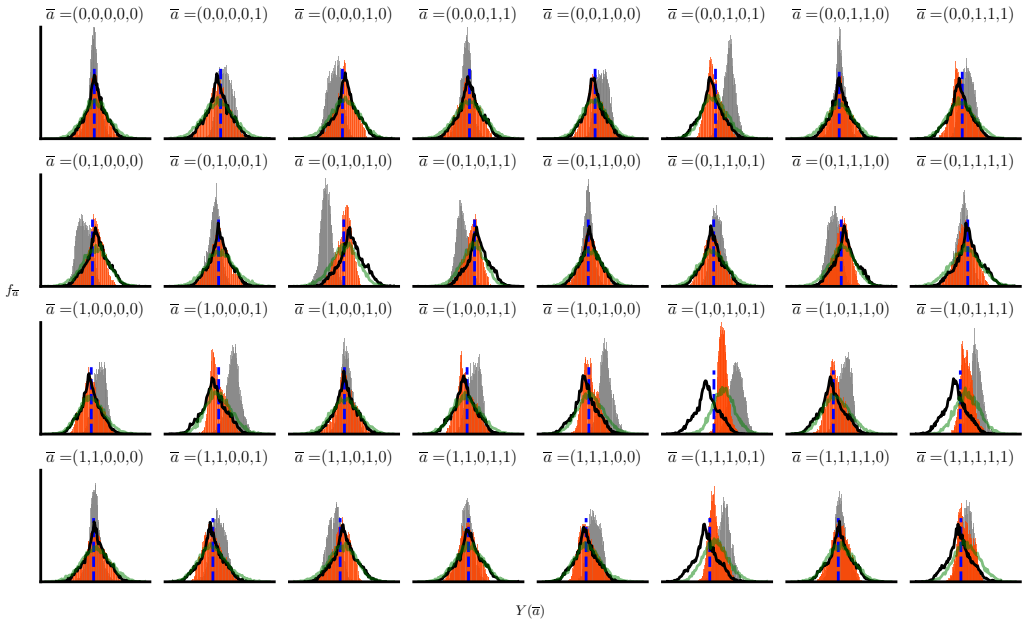
The real data used in this study comprises COVID-19-related demographic statistics collected from $3,219$ counties across 56 states/affiliated regions of the United States. The data covers a time period from 2020 to 2022. We obtained the data from reputable sources including the U.S. Census Bureau [9], the Center for Disease Control and Prevention [16], Google [19], the CMU DELPHI group's Facebook survey [21], and the New York Times [68]. To capture a relevant time window for analysis, we set the history dependence length d to 3, as most COVID-19 symptoms tend to subside within this timeframe [38].

In our analysis, the treatment variable A is the state-wise mask mandate indicator variable. A value of 0 indicates no mask mandate, while a value of 1 indicates the enforcement of a mask mandate. Notably, we observe a pattern in the data where mask mandates are typically implemented simultaneously across all counties within a state. This synchronization justifies the use of the state-wise mask mandate count as the treatment variable. As for the covariates X, we choose the county-wise incremental death count, state-wise the average retail and recreation metric (representing changes in mobility levels compared to a baseline, which can be negative), the state-wise symptom value, and the state-wise vaccine dosage.

Pennsylvania COVID-19 mask mandate data For the semi-synthetic dataset, we specifically look at the data within the state of Pennsylvania because of its long records spanning 106 weeks from 2020 to 2021. We set the four state-level covariates (per 100K people): the number of deaths, the average retail and recreation mobility, the surveyed COVID-19 symptoms, and the number of administered COVID-19 vaccine doses. We set the county-level incremental death count to the state level by computing a state average. We set the state-level mask mandate policy as the treatment variable, and the county-level number of new COVID-19 cases (per 100K) as the outcome variable, resulting in $m = 67$ since there are 67 counties in the state of Pennsylvania. We simulate $2,000$ trajectories of the (X, A, Y) tuples of 300 time points (each point corresponding to a week) according to the following formula:



(a) $d = 1$ (b) $d = 3$



(c) $d = 5$

Figure 10: The estimated and true counterfactual distributions across various lengths of history dependence ($d = 1, 3, 5$) on synthetic datasets with imbalanced proportions of different treatment ($\beta_0 = -2$). Each sub-panel provides a comparison for a specific treatment combination $\overline{a}$. We exclude `KDE` and `G-Net` for illustrative purposes.

$$X_0 \sim \text{Real-World}(\cdot), \tag{18}$$

$$X_t = \hat{\mathbb{P}}(X_t|\overline{A}_t, \overline{X}_t), \tag{19}$$

$$\mathbb{P}\{A_t = 1\} = \sigma(\beta_0 + \sum_{\tau=t-2}^{t} \beta_{t-\tau} A_\tau + \sum_{\tau=t-2}^{t} \beta_{t-\tau+3} X_\tau), \tag{20}$$

$$Y_t^{base} = -0.2A_{t-2} - 0.15A_{t-1} - 0.1A_t + 0.45 + \epsilon, \tag{21}$$

$$\mathbb{P}(L_t = 1) = \text{Bernoulli}(\prod_{j=1}^{4} X_\tau(j)), \tag{22}$$

$$Y_t(s) = Y_t^{base} + \begin{cases} \log(\mathcal{N}(s, \mu = [40.009, -75.133]^T, \Sigma = \mathbf{I})); \text{ if } L_t = 1, \\ \log(\mathcal{N}(s, \mu = [40.470, -79.980]^T, \Sigma = \mathbf{I})); \text{ otherwise.} \end{cases}, \tag{23}$$

where $\hat{\mathbb{P}}(\cdot)$ is learned with a 2-layer fully-connected neural network using the real data, $\epsilon \sim \mathcal{N}(0, 0.001)$ is the observation noise, s is the 2-dimensional coordinate of a entry (county) in Y_t, $\sigma(\cdot)$ is a Sigmoid function. All other coefficients are set according to Table 2 to ensure the generation of diverse data distributions. We generate the counterfactual samples according to Algorithm 2 by replacing the corresponding outcome models with the formulations above. In summary, the hotspot (mode of the Y_t vector) is either Philadelphia ($L_t = 1$) or Pittsburgh ($L_t = 0$), where the probability depends on the covariates $\overline{X}_t$. The values in the entries of Y_t follow the log-likelihood of a 2-dimensional isotropic Gaussian centered at the hotspot. As a result, the counterfactual and observed distributions will be bimodally distributed with different hotspot probabilities. We can then visually assess the performance of the models by comparing the distribution of the hotspot from the generated outcome samples to those of the counterfactual samples, as in Figure. 5b.

Nationwide COVID-19 Mask data We perform a case study using real data by looking at the aggregated COVID-19 data sources from 2020 to 2021 spanning 49 weeks due to the limited availability of the nationwide data. We exclude 89 counties with zero incremental new cases count. These counties either do not have a significant amount of infectious cases or have small populations, leading to $3,130$ counties across 56 states/affiliated regions of the United States. For variables that only have state-level records, we map them to the county level for simplicity.

We analyze the same set of variables as the semi-synthetic COVID-19 dataset but exclude the vaccine dosage covariate because of missing data in some states. To align the outcome variable with the covariates and treatment, we set it to measure one week after these variables. Due to the long-tailed distribution of the outcome variable, we apply a base-10 logarithmic transformation during the modeling process. Further details regarding the variables can be found in Table 3. We use the same model architecture described in Appendix G.3 to train the IPTW network and the `MSCVAE`. We generate counterfactual outcomes for treatment combinations $\overline{a} = (0, 0, 0)$ and $\overline{a} = (1, 1, 1)$. Since other treatment combinations occur rarely (less than 5% of observations), we exclude them from the final results.

H Additional synthetic results

In the main paper, we presented a visual comparison of the learned counterfactual distributions and the true counterfactual distribution for various scenarios ($d = 1, 3$), as shown in Figure 4. Here, in Figure 9 we show the case for $d = 5$. We also provide a similar comparison while setting $\beta_0 = -2$ (as opposed to $\beta_0 = -0.5$,) where the treatment combinations are imbalancedly distributed (Figure 10). Consistent with the findings in Figure 4, our results in Figures 9 and 10 demonstrate the superior performance of the `MSCVAE` model (represented by the orange shading) in accurately capturing the shape of the true counterfactual distributions (represented by the black line) across all scenarios. This observation further validates the quantitative comparisons presented in Table 1, where `MSCVAE` achieves the smallest mean and Wasserstein distance among all baseline methods. These results highlight that our algorithm attains competitive performance even when certain treatment combinations occur less frequently compared to others. This situation is common in real-life scenarios where certain treatment combinations are favored due to factors such as policy inertia.