Deep Counterfactual Estimation with Categorical Background Variables

Edward De Brouwer
ESAT-STADIUS
KU Leuven
edward.debrouwer@esat.kuleuven.be

Abstract

Referred to as the third rung of the causal inference ladder, counterfactual queries typically ask the "*What if ?*" question retrospectively. The standard approach to estimate counterfactuals resides in using a structural equation model that accurately reflects the underlying data generating process. However, such models are seldom available in practice and one usually wishes to infer them from observational data alone. Unfortunately, the correct structural equation model is in general not identifiable from the observed factual distribution. Nevertheless, in this work, we show that under the assumption that the main latent contributors to the treatment responses are categorical, the counterfactuals can be still reliably predicted. Building upon this assumption, we introduce CounterFactual Query Prediction (CFQP), a novel method to infer counterfactuals from continuous observations when the background variables are categorical. We show that our method significantly outperforms previously available deep-learning-based counterfactual methods, both theoretically and empirically on time series and image data. Our code is available at https://github.com/edebrouwer/cfqp.

1 Introduction

Counterfactual queries aim at inferring the impact of a treatment conditioned on another observed treatment outcome.Typically, given an individual, a treatment assignment, and a treatment outcome, the counterfactual question asks what would have happened to that individual, had it been given another treatment, everything else being equal. An illustrative and motivating example is the case of clinical time series. Based on the observation of the outcome of treatment A on a particular patient, counterfactual queries ask what would have been the outcome for this patient, had it been given treatment B instead. Notably, counterfactual prediction differs from interventional prediction, which is also referred to as counterfactual potential outcomes [22] and constitutes the second rung of the causation ladder [18]. Counterfactual predictions are retrospective, as they condition on an observed treatment outcome. In contrast, interventional predictions are prospective as they only condition on observations obtained before treatment assignment.

Much more than a statistical curiosity, counterfactual reasoning reflects complex cognitive abilities that are deeply ingrained in the human brain [21] and emerges in the early stages of cognitive development [7]. The ability to reason counterfactually can indeed help to identify causes of outcomes retrospectively, has been suggested to be central in the formation of rational intention, and supports key theories of human cognition [27]. The importance of counterfactual reasoning in the human cognitive process has thus motivated researchers to endow artificially intelligent systems with the same ability [20]. However, counterfactual inference is not possible from observational and interventional data alone [19].

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Counterfactual reasoning, therefore, requires making several assumptions to overcome this limitation. The most popular one assumes knowledge of the underlying structural equation model that describes the data generating process [20] or a specific functional form thereof [1, 5, 20]. Unfortunately, this assumption is rarely met in practice, especially in high-dimensional data such as time series or images. This led to the development of *deep* structural equation models that attempt to model the structural equations with neural networks [16, 23]. However, despite their ability to model high-dimensional data, these approaches fail to provide theoretical guarantees for the reconstruction of counterfactuals. Indeed, they focus on modeling the factual distribution which, without further assumption, can, unfortunately, lead to erroneous counterfactual distributions.

In this work, we bridge the gap between the classical structural equation model assumptions and deep-learning-based architectures. By assuming that the treatment and observables are continuous and that the hidden variables that contribute most to the treatment response are categorical, we show that we can approximately recover the counterfactuals using arbitrary parametric functions (*i.e.* deep neural networks) to model the causal dependence between variables. This allows us to infer counterfactuals on high-dimensional data such as time series and images. Generally, this work explores the assumptions that can lead to approximate counterfactual reconstructions while controlling the discrepancy between the recovered and the true counterfactual distributions.

Besides the general appeal of endowing machine learning architectures with counterfactual reasoning abilities, an important motivation for our work is the counterfactual estimation of treatment effects in clinical patient trajectories. In this motivational example, one wishes to predict the individual treatment effect retrospectively. Based on the observed treatment outcome of a particular patient, we want to predict what would have been the outcome under a different treatment assignment. The ability to perform counterfactual inference on patient trajectories has indeed been identified as a potential tool for improving long and costly randomized clinical trials [14].

Contributions

- We provide a new set of assumptions under which the counterfactuals are identifiable using arbitrary neural networks architecture, bridging the gap between structural equation models and deep learning architectures.
- We derive a new counterfactual identifiability result that motivates a novel counterfactual reconstruction architecture.
- We evaluate our construction on three different datasets with different high-dimensional modalities (images and time series) and demonstrate accurate counterfactual estimation.

2 Background

2.1 Problem Setup : Counterfactual Estimation

We consider the general causal model $M = \langle U, V, F \rangle$ depicted in Figure 1a consisting of background variables U, endogenous variables X, T and Y and the set of structural functions F. Background variables $U = \{U_X, U_T, U_\epsilon, W\}$ are hidden exogenous random variables that determine the values of the observed variables $V = \{X, T, Y\}$. Covariates $X \in \mathcal{X}$ represent the information available before treatment assignment, $T \in \mathcal{T}$ is the treatment assignment and $Y \in \mathcal{Y}$ is the observed response to the treatment. We refer to the space of probability measures on $\mathcal{Y}$ as $\mathcal{P}(\mathcal{Y})$. Observed variables V are generated following the structural equations $F = \{f_X, f_T, f_Y, f_\epsilon\}$, such that $X = f_X(U_x, W)$, $T = f_T(U_T, X)$, $U_\epsilon = f_\epsilon(W)$, $Y = f_Y(X, T, U_\epsilon)$. We further assume *strong ignorability* (*i.e.* no hidden confounders between T and Y).

Using notations introduced in Pearl et al. [20], we define the potential response of a variable Y to an action $do(T = t)$ for a particular realization of $U = u$ as $Y_t(u)$. Our goal is to predict the counterfactual response, for a new treatment assignment ($T = t'$), conditioned on an observed initial treatment response. That is, the probability of observing a different treatment response under treatment t', after observing treatment response y for covariate x and treatment t. The probability of observing counterfactual y' then writes:

$$P(Y_{t'} = y' \mid X = x, Y = y, T = t) = \frac{P(Y_{t'} = y', X = x, T = t, Y = y)}{P(X = x, T = t, Y = y)}$$
$$= \int_u P(Y_{t'}(u) = y') P(U = u \mid X = x, T = t, Y = y), \quad (1)$$

and we refer to the counterfactual probability measure as $\nu_{t'}(x, y, t)$. Equation 1 suggests a natural three step procedure for computing the probability of counterfactual. First, the *abduction* step infers the probability of U conditioned on the observed treatment outcomes, covariates and treatments: $P(U = u \mid X = x, T = t, Y = y)$. Second, in the *action* step, one sets the new treatment in the causal model ($do(T = t')$). Lastly, in the *prediction* step, one can propagate the values of $U = u$ and $T = t'$ in the causal graph, using F, to compute $P(Y_{t'}(u) = y')$.

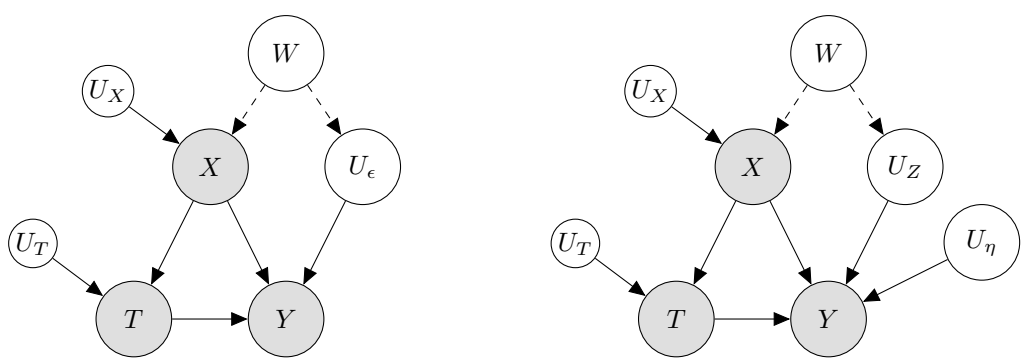
In practice, we only have access to a set of N observations of variables X, Y and T. We refer to this dataset as $\mathcal{D} = (\mathbf{X}, \mathbf{T}, \mathbf{Y})$ where $\mathbf{X} = \{x_i : i = 1, ..., N\}$, $\mathbf{Y} = \{y_i : i = 1, ..., n\}$ and $\mathbf{T} = \{t_i : i = 1, ..., N\}$. Importantly, we don't have access to counterfactual examples (*i.e.* a tuple (x, y, t, t', y')), such that direclty learning a map $(x, y, t, t') \to y_{t'}$ is excluded.

2.2 General Non-identifiability of Counterfactuals

Because the background variables U are hidden, the above three-steps procedure requires knowledge of the structural functions $F = \{f_X, f_T, f_Y\}$. Indeed, one can show that there exist multiple structural functions F that would lead to the same observed joint probability $P(X, Y, T)$ but would lead to incorrect counterfactual probabilities [18, 19]. The correct causal model is thus in general non-identifiable, leading to non-identifiability of the counterfactual probability. We specify what is meant by the identifiability of counterfactuals in the following definition.

Definition 1 (Identifiability of Counterfactuals)**.** Let ρ be a metric on $\mathcal{P}(\mathcal{Y})$, $\nu_{t'}(X, Y, T)$ the true counterfactual probability measure and $\hat{\nu}_{t'}(X, Y, T)$ the estimator of the counterfactual probability measure with N data points. Counterfactuals are ρ-identifiable at threshold δ if, for all $t, t' \in \mathcal{T}, x \in \mathcal{X}, y \in \mathcal{Y}$,

$$\lim_{N \to \infty} \rho(\nu_{t'}(X, Y, T), \hat{\nu}_{t'}(X, Y, T)) \leq \delta$$



(a) General Bayesian network for the treatment counterfactual problem. X, Y and T are observed while U_X, U_T, W and U_ϵ are hidden background variables.

(b) Bayesian Network embodying the hidden categorical background variable assumption. U_ϵ is split in a background categorical variable U_Z and a continuous background variable U_η.

Figure 1: Graphical model representations of the causal model M. We assume strong ignorability and continuous treatments T, observables X and responses Y.

2.3 Causal Model Assumptions for Counterfactual Idenfiability

Despite the general non-identifiability of structural equation models laid out above, we propose plausible assumptions that one can build upon to identify counterfactuals reliably. We first assume X and Y are continuous (potentially high dimensional) variables (such as images or time series). The treatment assignment T is also assumed continuous, and $\mathcal{X} \times \mathcal{T}$ is a connected space. Our first central assumption posits that the hidden variable U_ϵ factorizes into a categorical and a continuous variable.

Assumption 1 (Categorical Background Variable)**.** The background variable U_ϵ decomposes into a categorical latent variable $U_Z \in [K] = \{1, .., K\}$ and an independent exogenous continuous variable U_η.

This assumption is depicted in the graphical model of Figure 1b and embodies the intuition of different hidden groups that drive the treatment response. For instance, a treatment could have different responses depending on the stage of the disease a patient finds themself in. The disease stage is unobserved yet correlated with the observed covariates X (through W).

Due to the categorical nature of variable U_Z, one can write the conditional probability of Y as a mixture model:

$$P(Y = y \mid X = x, T = t) = \sum_{u_z \in \{1,..,K\}} P(U_Z = u_z) \cdot \int P(U_\eta = u_\eta) \mathbb{I}[f_Y(x, t, u_z, u_\eta) = y] du_\eta$$

We define γ as the probability density function of the conditional treatment response generated by f_Y, U_Z and U_η. γ is thus a mixture probability density function with mixture components[1] $\gamma_k \in \mathcal{P}(\mathcal{Y})$ and weights ω_k.

$$Y \mid X, T \sim \gamma(X, T) = \sum_{k=1}^{K} \omega_k \gamma_k(X, T) \tag{2}$$

Without loss of generality, we assume that $U_\eta \sim \mathcal{N}(\mathbf{0}, \Sigma^2)$. In the case of additive noise, the conditional distribution of Y becomes a mixture of Gaussians: $\gamma(X, T) = \sum_{k=1}^{K} \omega_k \mathcal{N}(\mu_k(X, T), \Sigma_k^2)$, where μ_k are functions mapping X and T to the mean of the mixture components and we consider different variances for each k. We now proceed with the next assumptions.

Assumption 2 (Continuity)**.** The moments of the probability density functions $\gamma_k(x, t)$ exist and are continuous functions of X and T:

$$\mu_k^r(x, t) = \mathbb{E}_{Y \sim \gamma_k(x,t)} [Y^r] \in C(x, t) \quad \forall r \in \mathbb{N}, k \in [K] \tag{3}$$

Assumption 3 (Clusterability)**.** For each $(x, t) \in (\mathcal{X}, \mathcal{T})$, the density $\gamma(x, t)$ is clusterable and the expected deviation of each $\gamma_k(x, t)$ is bounded by a constant $\delta \in \mathbb{R}$. That is, $\forall k \in [K]$, $\forall x, t \in (\mathcal{X} \times \mathcal{T})$, with $\mu_k(x, t) = \mathbb{E}_{Y \sim \gamma_k(x,t)}[Y]$:

$$\mathbb{E}_{Y \sim \gamma_k(x,t)} \left[\|Y - \mu_k(x, t)\|_2 \right] \leq \delta \tag{4}$$

In our motivating clinical example, Assumption 2 reflects that the probability of specific treatment response changes continuously over the set of observed covariates and treatments. In particular, the expected treatment outcome for a particular patient varies continuously with the treatment assignment, which is a common assumption, *e.g.* in clinical practice [13].

[1] The mixture components are defined such that for any subset $\mathcal{A} \subset \mathcal{Y}$, we have $\int_{\mathcal{A}} \gamma_k(X, T)(y) dy = \int_{\infty}^{\infty} \mathbb{I}[f_Y(X, T, U_Z = k, U_\eta) \in \mathcal{A}] dP(U_\eta)$. The mixutre weights are defined as $\alpha_k = P(U_Z = k)$.

Assumption 3 posits *clusterability* of the mixture components γ_k for which a rigorous mathematical definition is given in Appendix C. Intuitively, it supposes that patients with the same observed covariates and treatment assignment but different hidden group will show different treatment outcomes. We also bound the expected deviation of the mixture components γ_k that characterize the inter-group variability in the treatment response for a particular patient and treatment outcome.

3 Methods

3.1 Identifiability and Counterfactuals Reconstruction

For a fixed point $(X = x, T = t)$, Equation 2 is a finite mixture model, for which identifiability results are available [3]. Notably, these results guarantee identifiability up to a permutation of the latent class assignment $\sigma(\cdot) : [K] \to [K]$. That is, there exists some permutation $\sigma(\cdot) : [K] \to [K]$ such that $\hat{\gamma}_{\sigma_k(k)(x,t)} \approx \gamma_k(x,t)$, $\hat{\omega}_{\sigma_k(k)} \approx \omega_k$, where $\hat{\gamma}$ and $\hat{\omega}$ are the estimated density functions and weights. However, it does not entail identifiability of the counterfactuals in the sense of definition 1. Indeed, the action step of the counterfactual strategy from Section 2.1 requires a consistent permutation σ across the whole domain $(\mathcal{X} \times \mathcal{T})$ in order to reuse the inferred class assignments $\hat{U}_Z$ at a specific point $(X = x, T = t)$ to predict the counterfactual at another point $(X = x, T = t')$ - with a different treatment assignment. Nevertheless, using the assumptions from the previous section, we can still ensure the identifiability of the counterfactuals as the following result confirms:

Result 3.1 (Identifiabilty of Counterfactuals with Categorical Background Variables)**.** *Let X, T and Y be continuous random variables generated according to the graphical model of Figure1b with the domain of X and T being connected. Let $W_1(\cdot,\cdot)$ be the first Wasserstein distance on $\mathcal{P}(\mathcal{Y})$, $\nu_{t'}(X,Y,T)$ the probability distribution of $Y_{t'} \mid X,Y,T$ and $\hat{\nu}^N_{t'}(X,Y,T)$ its estimator from N observed data points. If Assumptions 1, 2 and 3 hold, for each (x,t), the counterfactual distribution is W_1-identifiable at threshold δ:*

$$\lim_{N\to\infty} \mathbb{E}_{Y\sim\gamma(x,t)} \left[W_1(\nu_{t'}(x,Y,t), \hat{\nu}^N_{t'}(x,Y,t)) \right] \leq \delta$$

In the special case when the noise response is additive, we have

$$\lim_{N\to\infty} W_1(\nu_{t'}(X,Y,T), \hat{\nu}^N_{t'}(X,Y,T)) = 0$$

The proof is given in Appendix C. This result gives us a bound on the distance between the inferred and true counterfactual distributions in the asymptotic regime. Importantly, it does not restrict the dimension of $\mathcal{X}$ and $\mathcal{Y}$, and is thus valid on challenging data modalities such as time series or images.

Continuity of distribution and complexity The result above holds asymptotically in the number of available samples. In the additive Gaussian case, the sample complexity for learning a K-mixture model with $Y \in \mathbb{R}^d$ within ϵ total variation distance is $\tilde{O}(Kd^2/\epsilon)$ [4]. Fortunately, the continuity assumption (Assumption 2) saves us from having to learn an individual mixture at each point $(X = x, T = t)$, by jointly learning the continuous moments functions $\mu'_r(X,T)$. A better sample complexity bound can then be derived with further assumptions on $\mu'_r(X,T)$.

3.2 CFQP : CounterFactual Query Prediction

Equipped with those theoretical results, we introduce CFQP, a counterfactual prediction model based on a neural Expectation-Maximization mechanism. The basic building block of CFQP is a base-model $m(x,t)$, that predicts the treatment response y based on covariates and treatment assignment. For each latent category k, we learn a base-model that approximates the individual treatment response in that category : $m_k(x,t) \approx \mu_k(x,t)$. Our theoretical results require the number of classes of K to be known in advance. Yet, this is rarely the case in practice, and we describe our architecture for an arbitrary number of classes K'. The learning of the base-models follows three steps: a joint initialization, an expectation phase, and a maximization phase. The overall process is depicted in Figure 2. We also present a pseudo-code description of the procedure in Algorithm 1.

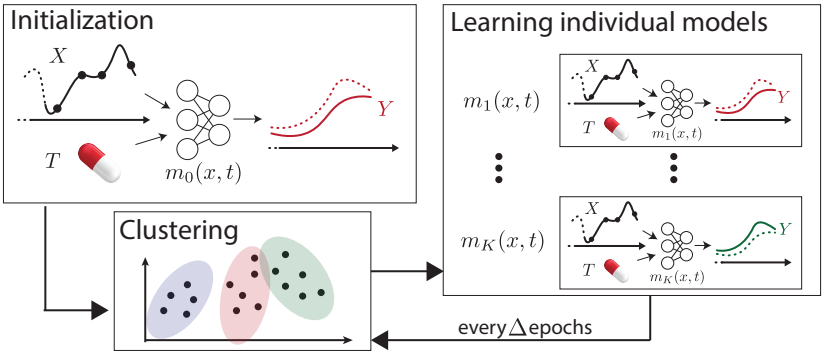


Figure 2: Training procedure of CFQP. We first initialize a common model m_0. Second, we cluster the different available data points into K' clusters. We then use this clustering to train K' individual models (Maximization step). The clustering is updated every Δ epochs (Expectation step).

Initialization We first train a single common base-model m_0 on all data points $\mathbf{X}, \mathbf{T}$ and $\mathbf{Y}$. Because it discards the variability of the hidden variable U_Z, this model will approximate the conditional average treatment response : $m_0(x,t) \approx \mathbb{E}[Y \mid X = x, T = t]$.

Expectation - Clustering The clustering step ensures that each base-model m_k is trained with the data from the corresponding latent category. We distinguish between an *initial* clustering and an *update* clustering stage.

The *initial* clustering assignment happens after the initialization step has converged. We then use the clusterability assumption (Assumption 3) to drive the assignment and use K'-means clustering based on the residuals $\|y - m_0(x,t)\|_2^2$.

The *update* clustering step happens at regular intervals to improve the quality of the cluster assignment as the performance of each individual base-model increases. A point (x, y, t) is assigned to cluster $k = \arg\min_j \|y - m_j(x,t)\|_2^2$.

Maximization - Learning individual base-models Once the dataset $\mathcal{D} = (\mathbf{X}, \mathbf{T}, \mathbf{Y})$ is clustered in K' groups, we train the individual models m_k on the corresponding cluster. This corresponds to the maximization step of the expectation-maximization scheme. Every Δ epochs, we update the cluster assignment until convergence.

Counterfactual prediction For a data point (x, t, y), we first infer the cluster assignment $k = \arg\min_j \|y - m_j(x,t)\|_2^2$. We predict the counterfactual for treatment t' as $\hat{y}' = m_k(x, t')$.

4 Related Work

Causal perspectives in machine learning have gained significant traction in the past years [25, 24]. Among them, one distinguishes between causal *discovery* approaches, aiming at discovering the causal relations between variables [9, 8], and causal *inference*, aiming at building treatment effects estimators from data [17]. Our work belongs to the latter. A common way to estimate counterfactuals is to posit a specific structural causal model [19] or its specific functional form [5], therefore ensuring identifiability. Synthetic controls are an example of this strategy, assuming an underlying linear structure [1]. However, these approaches are by definition restrictive in terms of the expressivity of the structural equations. Deep learning approaches for counterfactual estimation have been recently proposed to address this issue [16, 23] but without identifiability guarantee. For the discrete case, assumptions have been proposed to bridge this gap, such as monotonicity or generalization thereof [15, 12]. Nevertheless, because they focus on addressing the discrete case, they are not directly applicable to high-dimensional data such as time series, which is a motivation for our work.

Algorithm 1: CFQP Training

Data: $\mathbf{X}$,$\mathbf{Y}$,$\mathbf{T}$, the number of latent clusters K', a number of epochs e_{max}, the update-period Δ.
Result: A list of K' hidden models m_k.
Initialize a single base model m_0 at random.
for *epoch* $\leftarrow 0$ **to** e_{max} **do**
  Train m_0 minimizing $\mathcal{L}_0 = \mathbb{E}\left[(m_0(X,T) - Y)^2\right]$
end
Compute residuals $\mathbf{r} = (m_0(\mathbf{X},\mathbf{T}) - \mathbf{Y})$.
Assign the residuals of each training sample into K' clusters (initial).
Initialize each m_k with m_0.
for *epoch* $\leftarrow 0$ **to** e_{max} **do**
  for $i \leftarrow 0$ **to** Δ **do**
    Train m_k minimizing $\mathcal{L}_k = \mathbb{E}\left[(m_k(X_k,T_k) - Y_k)^2\right]$ for each cluster.
  end
  Compute residuals $\mathcal{R} = \{(m_k\left(\mathbf{X},\mathbf{T}\right) - \mathbf{Y}) : k = 1, ..., K'\}$.
  Assign the residuals of each training sample into K' clusters (update).
end
Return trained models m_k.

5 Experiments

5.1 Datasets

Treatment outcomes are often complex and high-dimensional. We thus evaluate our proposed model architecture on high-dimensional data modalities: time series and images. As counterfactual inference evaluation on real-world data is a complex and ongoing research area, we use synthetic datasets inspired by case studies from the literature. Details about the data generation are given in Appendix A.

5.1.1 Counterfactual Image Transformation

To explore the capacity of our model to operate on images, we generate a dataset inspired by [16], building upon modification of MNIST images. Covariates X are original MNIST images and the treatment T is a spatial rotation applied to the image. U_Z is a coloring of the image and the treatment outcome Y is the resulting colored and rotated MNIST image. U_η is either an additive Gaussian noise at each pixel or a Gaussian Blur with a random kernel.

5.1.2 Counterfactual Longitudinal Treatment Effect Prediction

Next, we explore the performance of our model on time series data. As per our motivational example, our goal is here to infer counterfactuals from clinical trajectories with a hidden patient class. X and Y are multi-dimensional time series, U_Z is a latent patient group and T is a continuous treatment assignment. We used two different datasets: an Harmonic Oscillator and a Cardiovascular model.

Harmonic oscillator We simulate the angular positions of two coupled harmonic oscillators. The treatment consists of applying different gradual offsets to the time series. We consider three hidden groups ($K = 3$) that modulate the treatment response differently.

Cardiovascular simulator We use a simulator of the cardiovascular system proposed in [28] to predict the impact of fluid intake on blood pressure. Fluids are commonly administered in intensive care for treating severe hypotension. However, the individual patient response is difficult to assess a priori, as it depends on clinically hidden variables. We consider here two hidden groups of patients with distinct responses to fluid intake. X and Y contain the arterial and venous blood pressure time series before and after fluid intake. The treatment is the amount of fluid injected. We add process noise by either considering an additive Gaussian noise on the observed treatment response or by introducing noise in the fluid injection process, leading to a non-linear and complex perturbation.

5.2 Baselines

We compare our approach against multiple counterfactual inference models from the literature: *Diff-SCM* [23], a recently introduced diffusion based counterfactual model; *Deep-SCM* [16], a deep counterfactual inference architecture building on normalizing flows ; Synthetic Controls (*SC*) [1], a well known linear method for counterfactuals estimation and *Deep-ITE*, a deep individual treatment effect estimator such as proposed in [26, 11], using direct predictions from X and T.

Table 1: Test MSE of the counterfactual reconstructions for the different datasets.

	Additive Noise			Non-Additive Noise		
Model	Harmonic Oscillator	colored-MNIST	Cardiovascular	Harmonic Oscillator	colored-MNIST	Cardiovascular
Deep-ITE [11]	0.187 ± 0.006	0.017 ± 0.001	1.084 ± 0.087	0.174 ± 0.004	0.017 ± 0.001	1.14 ± 0.121
SC [1]	0.177 ± 0.131	0.020 ± 0.001	1.610 ± 0.141	0.167 ± 0.004	0.020 ± 0.001	1.628 ± 0.144
Deep-SCM [16]	0.124 ± 0.005	0.011 ± 0.001	0.405 ± 0.042	0.123 ± 0.005	0.011 ± 0.001	0.424 ± 0.042
Diff-SCM [23]	0.082 ± 0.023	0.008 ± 0.004	0.206 ± 0.036	0.106 ± 0.038	0.009 ± 0.002	0.311 ± 0.073
CFQP (ours)	$\mathbf{0.013 \pm 0.001}$	$\mathbf{0.001 \pm 0.001}$	$\mathbf{0.077 \pm 0.050}$	$\mathbf{0.009 \pm 0.001}$	$\mathbf{0.002 \pm 0.001}$	$\mathbf{0.188 \pm 0.114}$

5.3 Counterfactuals Prediction

We evaluate our approach in terms of the quality of the counterfactual reconstructions. We generate counterfactuals (t',y') for each observation (x,y,t) in the test set, resulting in a tuple (x,y,t,t',y'). We evaluate the MSE between the true counterfactual y' and the estimated one $\hat{y}'$ as $\text{MSE}_{cf} = \|y' - \hat{y}'\|_2^2$. We set the number of clusters K as an hyper-parameter and select the value that result in lowest validation error. The results are presented in Table 1. We use additive and non-additive noise variants for each dataset. For each dataset and each variant, we see that our approach outperforms the other baselines, demonstrating its experimental effectiveness.

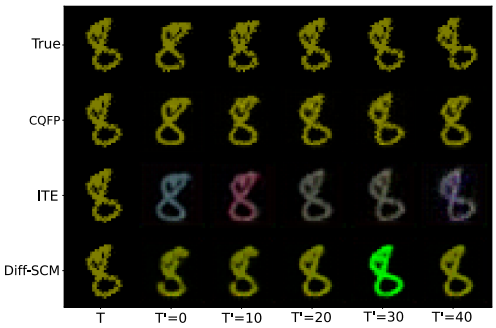


Figure 3: Example of reconstructed counterfactuals on the image dataset for different methods (best seen in color). Here, U_Z corresponds to the color of the digit and T to the rotation angle. X is the non-rotated and non-colored image while Y_T is the colored and rotated image. First column is the observed factual treatment outcome. Subsequent columns show the counterfactual reconstructions for different values of T'. First row is the ground truth.

In Figure 3, we also report examples of counterfactual reconstructions on the image dataset. Each row corresponds to counterfactual estimations for a particular method (top is the truth), and each column represents a different treatment assignment T' (the left column is the factual T). Because it cannot model the categorical latent variable, the Deep-ITE model gets the expected individual treatment effect right but fails to accurately estimate the counterfactuals. For Diff-SCM, we observe that, despite providing an accurate reconstruction for most of the treatment assignments, the lack of guarantees on the deep structural equation model leads to potential incorrect reconstructions (*e.g.* at $T' = 30$). To assess the quality of the counterfactual images reconstructions, we also evaluate the structural similarity index and report the results in Appendix A.1.1. We observe that CFQP outperforms all baselines on that metric.

5.4 Robustness Analysis

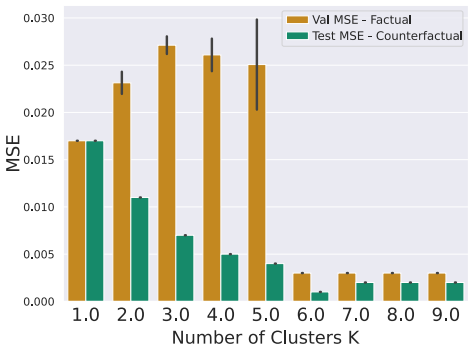
We further complement our experimental section with a comprehensive robustness analysis. We first investigate the impact of misspecifying the number of latent classes (*i.e.* when $K \neq K'$). We then investigate the impact of a correlation between X and Z with increasing strength. Finally, we investigate the robustness of our approach to different noise responses.

5.4.1 Number of latent classes

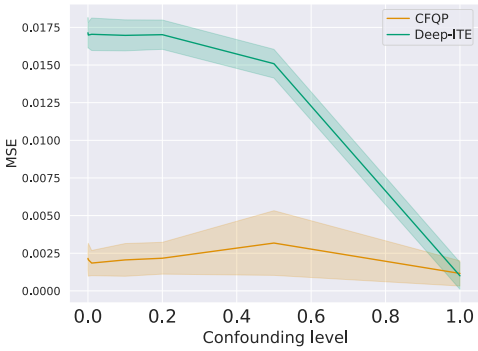
We study the evolution of performance when the number of classes K' is misspecified ($K \neq K'$). In Figure 4a, we show the reconstruction error on the validation set (*i.e.*, on the factual data) and the counterfactual reconstruction error of CFQP for different values of K' on the image dataset with true number of latent classes $K = 6$. The validation MSE is lowest for $K' = 6$ and $K' = 7$, which corresponds to optimal test reconstruction error. This hyper-parameter can thus easily be tuned by monitoring the treatment prediction performance on the factual data. In Appendix B.2, we further report the quantitative results for the other datasets in function of the number of the clusters K.

5.4.2 Correlation between X and U_Z

Our approach holds even when the latent background variable U_Z and the covariates X are correlated. When this is the case, information from X can be used to infer the value of U_Z. In Figure 4b, we study the impact of the correlation strength ρ between X and U_Z. More details about the definition of ρ are to be found in Appendix B. We compare the performance of CFQP with a Deep-ITE model. We observe that the counterfactual performance of our model remains constant, regardless of the correlation strength. However, as the correlation becomes more important, U_Z becomes a deterministic outcome of X and Deep-ITE eventually converges to the performance of CFQP.



(a) Reconstruction MSE in function of the number of groups. The true number of groups is 6.



(b) Evolution of the reconstruction MSE in function of the correlation strength between X and U_Z.

Figure 4: Robustness analysis for the number of latent classes (left) and correlation strength (right).

5.4.3 Non-additive Noise Responses

Our model architecture is motivated by the additive noise case. Yet, our theoretical results hold for arbitrary distributions. In Table 1, we reports the performance of the different models under non-additive noise responses. We observe an overall small degradation of performance of all methods but CFQP still outperforms baselines, as predicted by our identifiability results.

Table 2: Fluids PEHE

Model	Cardiovascular PEHE ↓
Deep-ITE [11]	0.594 ± 0.160
SC [1]	0.258 ± 0.016
Diff-SCM [23]	0.377 ± 0.049
CFQP (ours)	$\mathbf{0.143 \pm 0.108}$

5.5 Retrospective Individualized Treatment Effect Estimation

We explore the ability of our model to infer individualized treatment effects retrospectively. Based on observed treatment outcomes, we predict the difference in treatment response between two treatment regimes for a single individual. The metric of interest is the Precision in Estimation of Heterogeneous Effects PEHE [10]. For two treatment regimes t' and t'', we write

$$\text{PEHE}(t', t'') = \sqrt{\mathbb{E}_{X,T,Y}\left[\left((Y_{t''} - Y_{t'}) - (\hat{Y}_{t''} - \hat{Y}_{t'})\right)^2 \mid X, Y, T\right]}. \tag{5}$$

In Table 2, we report PEHE$(t' = 0.5, t'' = 0.8)$ for test patients in the Cardiovascular dataset with different fluid intake. We observe that the methods leveraging information about treatment outcomes, CFQP, SC and Diff-SCM outperform the classical individual treatment effect estimators (Deep-ITE).

5.5.1 Impact of the clustering algorithm

Our approach relies on an *initial* clustering step as described in Section 3.2. Different clustering strategies can be considered for this step. In Appendix B.5, we investigate the difference between K-means and Gaussian mixture models. Only minor deviations are observed in terms of performance between both approaches.

6 Conclusion

Estimating counterfactuals from observational studies is one of the most challenging tasks in causal inference. In this work, we proposed a set of reasonable assumptions that allow computing counterfactuals on high-dimensional data while harnessing the power of modern machine learning architectures. Based on these assumptions, we derived a new counterfactual model and demonstrated favorable experimental performance. In particular, we showed that this approach could be used to infer individual treatment effects a posteriori in clinical patient trajectories. Nevertheless, the set of assumptions proposed here is not unique, and depending on the specific applications, others might be deemed more relevant. Indeed, the space of assumptions that allow to approximately recover counterfactuals with a controlled level of error is a potentially very rich research direction. Other model architectures compatible with our set of assumptions are also possible.

Potential negative societal impacts Counterfactual inference has been used in clinical settings and for assessing the impact of public policies [2], among others. However, as shown in this paper, its correctness hinges on assumptions whose validity is not always met, potentially leading to misleading conclusions.

Acknowledgements Edward is funded by a FWO-SB grant. Edward would also like to thank Zeshan Hussain, David Sontag and Adam Arany for their comments.

References

[1] A. Abadie. Using synthetic controls: Feasibility, data requirements, and methodological aspects. *Journal of Economic Literature*, 59(2):391–425, 2021.

[2] A. Abadie, A. Diamond, and J. Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of california's tobacco control program. *Journal of the American statistical Association*, 105(490):493–505, 2010.

[3] B. Aragam, C. Dan, E. P. Xing, and P. Ravikumar. Identifiability of nonparametric mixture models and bayes optimal clustering. *The Annals of Statistics*, 48(4):2277–2302, 2020.

[4] H. Ashtiani, S. Ben-David, N. Harvey, C. Liaw, A. Mehrabian, and Y. Plan. Nearly tight sample complexity bounds for learning mixtures of gaussians via sample compression schemes. *Advances in Neural Information Processing Systems*, 31, 2018.

[5] A. Balke and J. Pearl. Counterfactuals and policy analysis in structural models. *arXiv preprint arXiv:1302.4929*, 2013.

[6] E. D. Brouwer, J. González, and S. Hyland. Predicting the impact of treatments over time with uncertainty aware neural differential equations. In *AISTATS 2022*, February 2022.

[7] R. M. Byrne. Counterfactual thought. *Annual review of psychology*, 67:135–157, 2016.

[8] E. De Brouwer, A. Arany, J. Simm, and Y. Moreau. Latent convergent cross mapping. In *International Conference on Learning Representations*, 2020.

[9] C. Glymour, K. Zhang, and P. Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019.

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

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

[12] G. Lorberbom, D. D. Johnson, C. J. Maddison, D. Tarlow, and T. Hazan. Learning generalized gumbel-max causal mechanisms. *Advances in Neural Information Processing Systems*, 34: 26792–26803, 2021.

[13] X. Monnet, P. E. Marik, and J.-L. Teboul. Prediction of fluid responsiveness: an update. *Annals of intensive care*, 6(1):1–11, 2016.

[14] T.-L. Nguyen, G. S. Collins, P. Landais, and Y. Le Manach. Counterfactual clinical prediction models could help to infer individualized treatment effects in randomized controlled trials—an illustration with the international stroke trial. *Journal of clinical epidemiology*, 125:47–56, 2020.

[15] M. Oberst and D. Sontag. Counterfactual off-policy evaluation with gumbel-max structural causal models. In *International Conference on Machine Learning*, pages 4881–4890. PMLR, 2019.

[16] N. Pawlowski, D. Coelho de Castro, and B. Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems*, 33:857–869, 2020.

[17] J. Pearl. Statistics and causal inference: A review. *Test*, 12(2):281–345, 2003.

[18] J. Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[19] J. Pearl. 7.1 causal and counterfactual inference. *The Handbook of Rationality*, page 427, 2021.

[20] J. Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19:2, 2000.

[21] N. J. Roese and J. M. Olson. *What might have been: The social psychology of counterfactual thinking*. Psychology Press, 2014.

[22] D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

[23] P. Sanchez and S. A. Tsaftaris. Diffusion causal models for counterfactual estimation. In *First Conference on Causal Learning and Reasoning*, 2021.

[24] B. Schölkopf. Causality for machine learning. In *Probabilistic and Causal Inference: The Works of Judea Pearl*, pages 765–804. 2022.

[25] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

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

[27] W. Starr. Counterfactuals. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Summer 2021 edition, 2021.

[28] S. Zenker, J. Rubin, and G. Clermont. From inverse problems in mathematical physiology to quantitative differential diagnoses. *PLoS Comput Biol*, 3(11):e204, 2007.

Checklist

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] The assumptions are cleary defined in section 2.3. The conclusion also relfects on the limitations of our work.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Last paragraph in the conclusion discusses the potential negative societal impacts of counterfactual inference. The main risk we identify is blind acceptance of the assumptions required for correct inference, which can lead to misleading conclusions in assessing treatment effects as posteriori.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] Our paper conforms to the ethics review guidlines.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The full set of assumptions is given in Section 2.3
   (b) Did you include complete proofs of all theoretical results? [Yes] The proofs of our results are given in Section C of the Appendix.
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] The code is available at the following url: `https://anonymous.4open.science/r/cfqp`
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide the training details in Section B of the Appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Error bars are reported from running the experiments 5 times (5 folds)
   (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] The total amount of compute and the type of resources used is available in the Appendix Section B
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] The authors of the code bases used for implementing the baselines are credited both in the code base and cited in the text. The list of software packages used is available at `https://anonymous.4open.science/r/cfqp/licenses.txt`
   (b) Did you mention the license of the assets? [Yes] The licenses of all software packages used in our experiments is available at `https://anonymous.4open.science/r/cfqp/licenses.txt`.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include an anonymized repository containing the code used for the experiments. It is available at `https://anonymous.4open.science/r/cfqp`
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We are not using any personal data.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] None of our datasets contains personal data.
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] No research with human subjects
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A] No research with human subjects
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A] No research with human subjects

A Datasets

A.1 Image Dataset Generation

The image dataset uses the MNIST handwritten digits images dataset. We use the following data generating process:

- $X \in \{0, ..., 255\}^{28\times28\times1}$ is an MNIST image sampled at random from the dataset. We write y_{label} the label of the digit present in the image.
- $T \in \mathcal{R} = \mathcal{U}(0, 0.3) + 5 \cdot \sigma\left(\frac{\frac{1}{784}\sum_{i=1}^{28}\sum_{j=1}^{28} X_{i,j} - 33}{11}\right)$ with $\sigma(\cdot)$ the sigmoid function. This encodes confounding between the observables X and the treatment assignment T.
- $p_0 = \frac{1}{K}$
- $p \in [0, 1]^K$
- $\rho \in [0, 1]$ is the strength of confounding between X and U_z
- $p[i] = \frac{1-((1-p_0)\rho+p_0)}{K-1} \quad \forall i \in [K], i \neq y_{label} \text{mod} K$
- $p[i] = ((1 - p_0)\rho + p_0) \quad \forall i \in [K], i = y_{label} \text{mod}(K)$
- $U_z \sim \mathcal{M}\text{ultinomial}(p)$
- $Y \in \{0, ..., 255\}^{28\times28\times1}$ is the rotated image X with angle T and Gaussian blur with kernel size $5\times$ and standard deviation σ.

Examples of generated images Y are shown in Figure 5. We use the original MNIST training set that we randomly divide as 70% training set, 15% validation set and 15% test set.

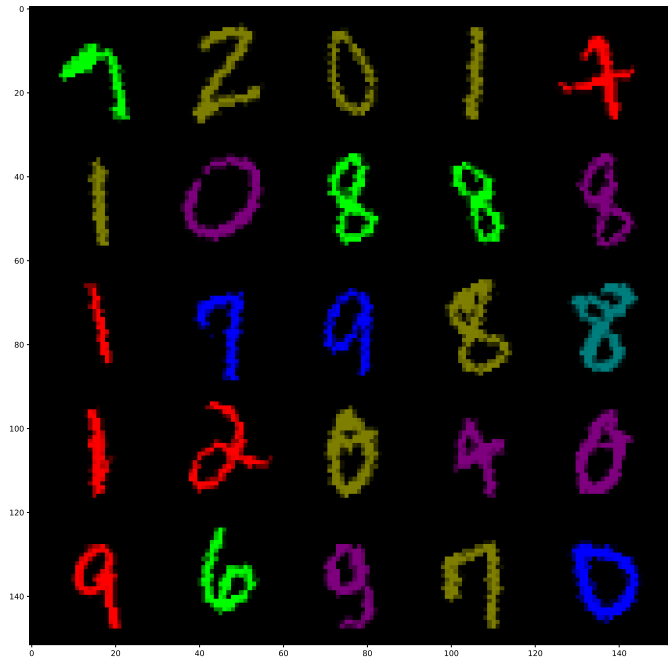


Figure 5: Examples of colored and rotated MNIST images Y

A.1.1 Evaluating the structural similarity index measure of the counterfactual images predictions

Mean square error between the ground truth and predictions might not always be the most meaningful way to assess the quality of the reconstruction. A widely used metric in computer vision is the structural similarity index measure (SSIM). In Table 3, we report the SSIM between the predicted and ground truth images counterfactuals. We observe that CFQPoutperforms all baselines in that metric, producing more accurate counterfactual predictions.

Table 3: Structural similarity results for the MNIST reconstructions

	Additive Noise	Non-Additive Noise
Model	colored-MNIST	colored-MNIST
Deep-ITE [11]	0.490 ± 0.006	0.484 ± 0.004
SC [1]	0.020 ± 0.001	0.020 ± 0.001
Deep-SCM [16]	0.857 ± 0.003	0.856 ± 0.004
Diff-SCM [23]	0.780 ± 0.063	0.750 ± 0.052
CFQP (ours)	0.964 ± 0.003	0.959 ± 0.005

A.2 Harmonic Oscillator

We consider input times series $X \in \mathbb{R}^{d_x \times t_x}$, with $d_x = 2$ the number of temporal dimensions and t_x the length of the input time series. The temporal responses are given by $Y \in \mathbb{R}^{d_y \times t_y}$. We use the following data generating process :

$$
\begin{aligned}
U_Z &\sim \{0, 1, 2\} \\
T &\sim \mathcal{U}(0.2, 1) \\
t_x &= (0, ..., 19) \\
t_y &= (20, ..., 40) \\
\phi &\sim \mathcal{N}(0, 1) \\
U_{\eta_x} &\sim \mathcal{N}(0, \sigma^2) \in \mathbb{R}^{t_x, 2} \\
U_{\eta_y} &\sim \mathcal{N}(0, \sigma^2) \in \mathbb{R}^{t_y, 2} \\
X &= [\sin(0.5t_x + \phi), \sin(0.5t_x + 2 * \phi] + \eta_x \\
Y &= [\sin(0.5t_y + \phi) + \Delta_0(T, t_y), sin(0.5 * t_y + 2 * \phi) + \Delta_1(T, t_y)] + \eta_y
\end{aligned}
$$

where $\Delta_0(T)$ and $\Delta_1(T)$ are defined as follows:

$$
\Delta_0(T, t) = \begin{cases} \frac{\min(t-20, t_p-20)}{t_p-20} \cdot T & \text{if } U_Z = 0 \text{ or } U_Z = 2 \\ 0 & \text{if } Z = 1 \end{cases}
$$

$$
\Delta_1(T, t) = \begin{cases} \frac{\min(t-20, t_p-20)}{t_p-20} \cdot T & \text{if } U_Z = 1 \text{ or } U_Z = 2 \\ 0 & \text{if } Z = 0 \end{cases}
$$

$t_p = 3$ characterizes the time constant of the treatment response dynamics. The latent categorical variables U_Z represent hidden patient groups that drive the treatment response. In the non-additive noise-response case, we set :

$$
\phi \sim \mathcal{N}(0, \sigma^2)^{40}
$$

such that the phase ϕ is time-varying and stochastic. Examples of the trajectories and reconstructed counterfactuals by CFQPare provided in Figure **??**.

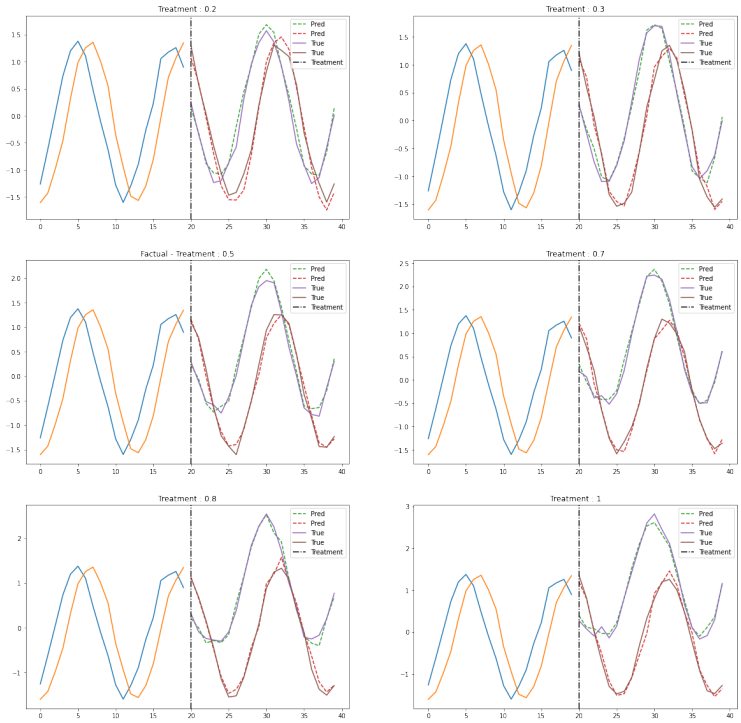


Figure 6: Counterfactuals generated by the model for different levels of treatment. The factual treatment is $T = 0.5$.

A.3 Cardiovascular Dataset

We use an ODE model of the cardiovascular system as proposed in [28, 6]. Fluid intake is commonly used for treating severe hypotension. However, the response of patient to fluids intake is difficult to assess beforehand. In particular, it depends on the patients cardiac contractility factor and the blood pressure at time of injection. If blood pressure is commonly and easily measured in standard clinical practice, assessing the cardiac contractility level of a patient requires imaging techinques such as echocardiography to measure the stroke volume. Yet, the injection of significant volume of fluids in an irreponsive patients cardiovascular system can lead to severe damage. This lead some clinicians to advocate for fluid challenges, or limited amount of fluid injection to test the responsiveness. This technique is still contested in the medical community and legs raising challenge, much less damaging but also less effective at assessing a patients response has been encouraged. This lack of availability of clear guidelines for fluids intake makes it a perfect case study for counterfactual prediction. Indeed, we'll try to address the question of if a clinician should administer fluids to particular patient based on his clinical history and therefore help informing clinical practice. The system of ODE used to generate the data is the following :

$$
\begin{aligned}
\frac{dSV(t)}{dt} &= I_{\text{external }(t)} \\
\frac{dP_a(t)}{dt} &= \frac{1}{C_a}\left(\frac{P_a(t) - P_v(t)}{R_{TPR}(S)} - SV \cdot f_{HR}(S)\right) \\
\frac{dP_v(t)}{dt} &= \frac{1}{C_v}\left(-C_a \frac{dP_a(t)}{dt} + I_{\text{external }(t)}\right) \\
\frac{dS(t)}{dt} &= \frac{1}{\tau_{\text{Baro}}}\left(1 - \frac{1}{1 + e^{-k_{\text{width}}\left(P_a(t) - P_{a_{\text{set}}}\right)}} - S\right)
\end{aligned}
$$

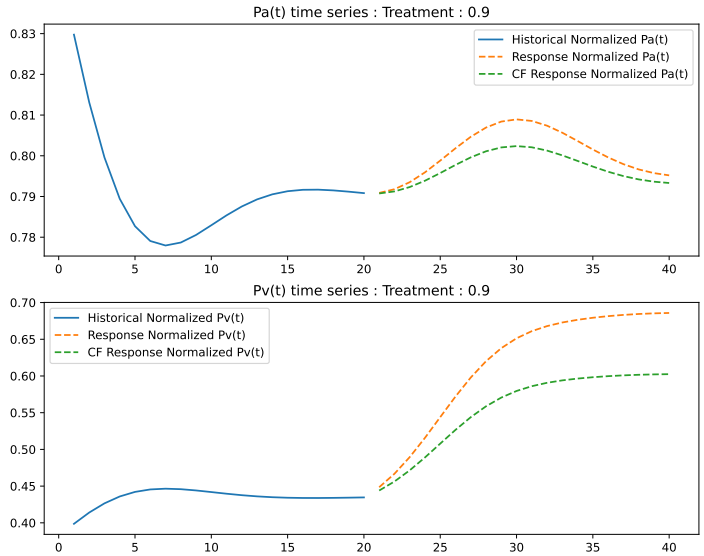


Figure 7: Example of time series in the cardio-vascular data set.

where

$$R_{TPR}(S(t)) = S(t)\left(R_{TPR_{Max}} - R_{TPR_{Min}}\right) + R_{TPR_{Min}} + R_{TPR_{Mod}}$$
$$f_{HR}(S(t)) = S(t)\left(f_{HR_{Max}} - f_{HR_{Min}}\right) + f_{HR_{Min}}.$$

In the above dynamical system, P_a, P_v, S and SV stand for arterial blood pressure, venous blood pressure, autonomic baroreflex tone and cardiac stroke volume respectively. $I_{\text{external }(t)}$ is the amount of fluids given the patient over time and corresponds to the exogeneous input $u_T(t)$ in our model. In the data generation, we model it as

$$I_{\text{external}}(t) = (1 + 2U_z) \cdot T \cdot 5 \cdot f(P_a(t=0)) \cdot e^{-(\frac{t - t_{\text{treat}} - 5}{5})^2}$$
$$I_{\text{external}}(t) = 0 \quad \forall t \leq t_treat$$

where the treatment assignment T is generated as $T \sim \mathcal{U}(0.6, 1)$ and the hidden group assignment $U_Z \sim \mathcal{B}\text{er}(0.5)$ and t_{treat} is the time of treatment. The function f introduces confounding in the treatment assignment by setting :

$$f(P_a(t=0)) = g(0.5 + (P_a(t=0) - 0.75)/0.1)g(x) \quad = 0.02 \cdot \left((cos)(5x - 0.2) \cdot (5 - x)^2\right)^2$$

We simulate the above system of ODEs for $t_{\text{span}} = 40$ seconds and sample an observation every $\Delta_t = 1$ second. The treatment assignment time is set to $t_{\text{treat}} = 20$.

We then set X as the first two dimensions of the dynamical system $((P_a, P_v)$ for $t < t_{\text{treat}}$ and Y as $((P_a, P_v)$ for $t \geq t_{\text{treat}}$. An example of a generated time series is given in Figure 7.

A.3.1 Noise Responses

Additive noise In the additive noise case, we add $U_\eta \sim \mathcal{N}(0, \sigma^2)^t_{\text{span}}$ on both X and Y

Non-additive noise In the non additive case, we modify the treatment assignment by setting:

$$I_{\text{external}}(t) = (1 + 2U_z + U_\eta(t)) \cdot T \cdot 5 \cdot f(P_a(t=0)) \cdot e^{-(\frac{t - t_{\text{treat}} - 5}{5})^2}$$

Where $U_{eta}(t)$ is a Gaussian process with zero-mean and diagonal covariance function with variance σ^2.

B Experiments Details

B.1 Computational Resources

We ran our experiments on cluster containing two types of GPUs : NVIDIA Titan Xp and NVIDIA Quadro GV100. Our experiments resulted in 7 GPU-days of computation.

B.1.1 Hyper-parameters

In Table 4, we report the hyper-parameters used in our experiments.

Hyper-parameter	Description	Image		Harmonic Oscillator		Cardiovascular	
		additive	non-additive	additive	non-additive	additive	non-additive
lr	learning rate	0.001	0.001	0.001	0.001	0.001	0.001
Δ	update period	10	10	20	20	20	20
σ	noise variance	0.01	0.05	0.05	0.05	0.01	0.01
epochs_0	number of epochs initialization	50	50	500	500	500	500
epochs_1	number of epochs fine-tune	50	50	500	500	500	500
K	number of classes	50	6	6	3	3	2
bs	batch size	128	128	128	128	128	128
N_{train}	Number of train samples	$42,000$	$42,000$	128	128	500	500
N_{val}	Number of train samples	$9,000$	$9,000$	128	128	250	250
N_{test}	Number of train samples	$1,000$	$1,000$	$1,000$	128	$1,000$	$1,000$

Table 4: Hyper-parameters used for training CFQP on the different datasets.

B.2 Number of groups

As discussed in section 5.4, the number of groups is an important hyper-parameter in our model. In Table 5,we report the validation reconstruction loss and the corresponding test counterfactual reconstruction MSE for all datasets and different values of K. We observe that we obtain a lower validation error for the true value of K in all datasets. In Figures 8a and 8a, we visualize the validation and test performance graphically for the Harmonic Oscillator dataset. We do the same on Figures 9a and 9b for the Cardiovascular dataset and on Figures 10a and 10b for the colored MNIST dataset.

B.3 Strength of Correlation Experiment

Our approach holds even when the latent background variable U_Z and the covariates X are correlated. When this is the case, information from X can be used to infer the value of U_Z. In Figure 4b, we show the impact of the correlation strength ρ between X and U_Z.

We incorporate correlation between X and U_z as follows. The base probability p_0^k of each U_z is uniform. That is,

$$P^0(U_z = k) = p_0^k = \frac{1}{K}$$

We modify this probabilty based on the label of the image (the digit label $y_{\text{label}} \in [9]$. For a correlation factor $\rho \in [0, 1]$, we write

Data	Noise	Number of Centers	Validation MSE	Test MSE
MNIST	Additive	1	0.0171 ± 0.00012	0.01657 ± 0.00038
MNIST	Additive	2	0.02267 ± 0.00056	0.01094 ± 0.00025
MNIST	Additive	3	0.02746 ± 0.00081	$0.00677 \pm 9e-05$
MNIST	Additive	4	0.02552 ± 0.0022	$0.00529 \pm 1e-04$
MNIST	Additive	5	0.02556 ± 0.01707	0.00377 ± 0.00016
MNIST	Additive	6	$\mathbf{0.00271 \pm 0.0003}$	$0.00145 \pm 6e-05$
MNIST	Additive	7	0.00272 ± 0.00028	$0.0016 \pm 7e-05$
MNIST	Additive	8	0.00278 ± 0.00027	$0.00167 \pm 7e-05$
MNIST	Additive	9	0.00289 ± 0.00031	$0.00179 \pm 6e-05$
MNIST	Non-Additive	1	$0.01673 \pm 9e-05$	0.01673 ± 0.00026
MNIST	Non-Additive	2	0.02228 ± 0.0004	0.01125 ± 0.00026
MNIST	Non-Additive	3	0.02632 ± 0.00063	0.00693 ± 0.00017
MNIST	Non-Additive	4	0.03182 ± 0.00653	0.00552 ± 0.00017
MNIST	Non-Additive	5	0.0041 ± 0.00018	0.004 ± 0.00018
MNIST	Non-Additive	6	$\mathbf{0.00179 \pm 0.00014}$	0.00171 ± 0.00012
MNIST	Non-Additive	7	0.00186 ± 0.00014	0.00185 ± 0.00011
MNIST	Non-Additive	8	0.00198 ± 0.00022	0.00198 ± 0.00018
MNIST	Non-Additive	9	0.00212 ± 0.00023	0.00217 ± 0.00022
Harmonic Oscillator	Additive	1	0.1653 ± 0.01086	0.16584 ± 0.00443
Harmonic Oscillator	Additive	2	0.23024 ± 0.04305	0.08058 ± 0.01735
Harmonic Oscillator	Additive	3	$\mathbf{9e-05 \pm 3e-05}$	0.01332 ± 0.00123
Harmonic Oscillator	Additive	4	$0.00013 \pm 1e-04$	0.0199 ± 0.00604
Harmonic Oscillator	Additive	5	0.00035 ± 0.00024	0.04017 ± 0.00697
Harmonic Oscillator	Non-additive	1	0.17451 ± 0.01053	0.17459 ± 0.00704
Harmonic Oscillator	Non-additive	2	0.15164 ± 0.10242	0.09669 ± 0.01678
Harmonic Oscillator	Non-additive	3	$\mathbf{0.00891 \pm 0.00035}$	0.00956 ± 0.00096
Harmonic Oscillator	Non-additive	4	0.00905 ± 0.00032	0.02279 ± 0.00327
Harmonic Oscillator	Non-additive	5	0.00984 ± 0.00073	0.03464 ± 0.01158
CardioVascular	Additive	1	1.01453 ± 0.06228	1.02569 ± 0.04794
CardioVascular	Additive	2	0.03677 ± 0.01694	0.07701 ± 0.05001
CardioVascular	Additive	3	0.04906 ± 0.02931	0.11363 ± 0.06759
CardioVascular	Non-additive	1	1.03865 ± 0.12068	1.09768 ± 0.05459
CardioVascular	Non-additive	2	$\mathbf{0.1372 \pm 0.05497}$	0.18797 ± 0.11421
CardioVascular	Non-additive	3	0.15747 ± 0.10191	0.2434 ± 0.13175

Table 5: Reconstruction validation MSE and counterfactual test MSE for all datasets and different number of groups (K). Lowest validation MSE are bolded for each dataset.

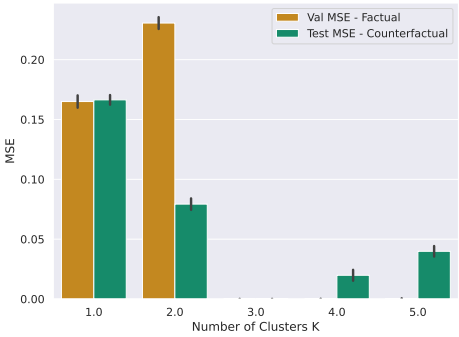
$$p^k = \begin{cases} \frac{1-((1-p_0)\rho+p_0)}{K-1} & \text{if } k \neq y_{label}\text{mod}(K) \\ ((1-p_0)\rho + p_0) & \text{if } k = y_{label}\text{mod}(K) \end{cases}$$

The above equation suggest that some images with specific label classes will have higher probability of sampling a latent class $U_Z = u_z$ such that $y_{\text{label}}\text{mod}(K) = u_z$.

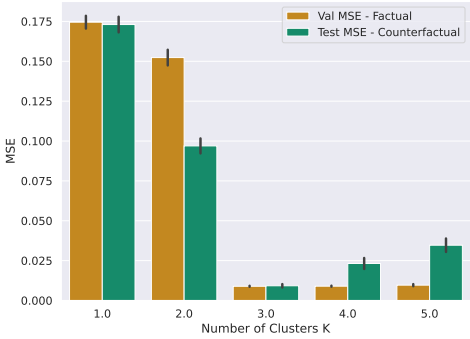
In the case when $\rho = 0$, $p^k = p_0^k = \frac{1}{K}$, corresponding to a uniform probability of latent class for each class label. When $\rho = 1$, $p^{y_{\text{label}}\text{mod}_{(K)}} = 1$ and the other probabilities are set to 0, corresponding to a deterministic assignment of U_Z conditioned on y_{label}.

B.4 Scalability

CFQPscales linearly with the number of groups K, though it can be easily parallelized. Indeed, each sub-model m_i can be evaluated in parallel. Regarding scaling with respect to the dimensions

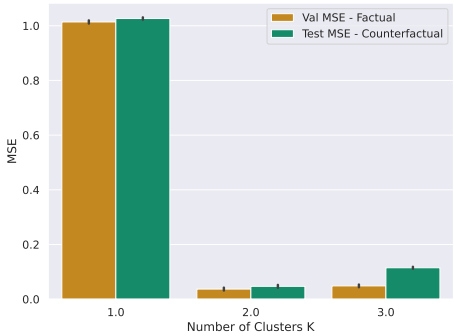


(a) Reconstruction MSE in function of the number of groups in the Harmonic Oscillator dataset. The true number of groups is 3.

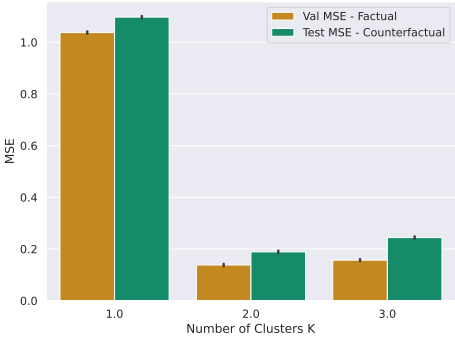


(b) Reconstruction MSE in function of the number of groups in the Harmonic Oscillator dataset. The true number of groups is 3.

Figure 8: Analysis of the impact of the number of clusters K on the validation and test performance for the Harmonic Oscillator dataset.



(a) Reconstruction MSE in function of the number of groups in the Cardiovascular dataset. The true number of groups is 2.



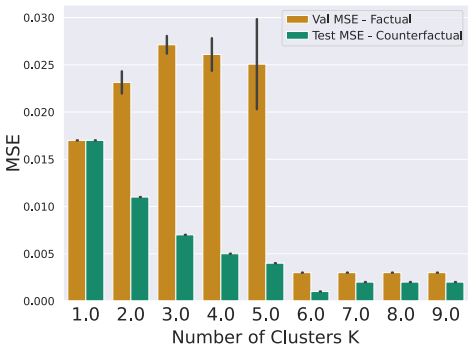
(b) Reconstruction MSE in function of the number of groups in the Cardiovascular dataset. The true number of groups is 2.

Figure 9: Analysis of the impact of the number of clusters K on the validation and test performance for the Cardiovascular dataset.

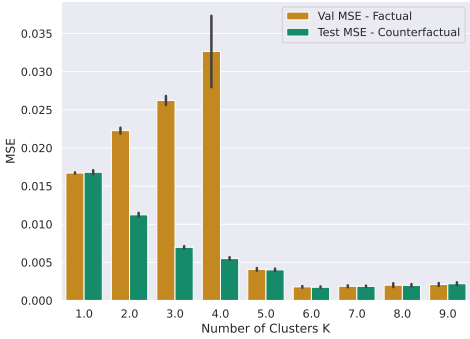
of the outcomes and treatments, we can use arbitrary neural networks architectures to process them, for which the complexity can vary. In our experiments, we used CNNs for images, which scale quadratically with the size of the images and TemporalCNNs for time series which scale linearly with the length of the time series as well as number dimensions.

B.5 Clustering Algorithm

Our approach relies on an *initial* clustering step as described in Section 3.2. In our main experiments, we use a k-means algorithm for this step. However, different clustering strategies can be considered. In Table 6, we report the performance metrics of a variant of our approach using Gaussian Mixture Models (GMM). For computational reasons, the GMM is fit by using a random sub-sample of the training data (1000 samples). We observe only minor performance deviations between k-means and GMM versions of CFQP.



(a) Reconstruction MSE in function of the number of groups in the colored MNIST dataset. The true number of groups is 6.



(b) Reconstruction MSE in function of the number of groups in the colored MNIST dataset. The true number of groups is 6.

Figure 10: Analysis of the impact of the number of clusters K on the validation and test performance for the colored MNIST dataset.

Table 6: Test MSE of the counterfactual reconstructions for the different datasets.

	Additive Noise			Non-Additive Noise		
Model	Harmonic Oscillator	colored-MNIST	Cardiovascular	Harmonic Oscillator	colored-MNIST	Cardiovascular
Deep-ITE [11]	0.187 ± 0.006	0.017 ± 0.001	1.084 ± 0.087	0.174 ± 0.004	0.017 ± 0.001	1.14 ± 0.121
SC [1]	0.177 ± 0.131	0.020 ± 0.001	1.610 ± 0.141	0.167 ± 0.004	0.020 ± 0.001	1.628 ± 0.144
Deep-SCM [16]	0.124 ± 0.005	0.011 ± 0.001	0.405 ± 0.042	0.123 ± 0.005	0.011 ± 0.001	0.424 ± 0.042
Diff-SCM [23]	0.082 ± 0.023	0.008 ± 0.004	0.206 ± 0.036	0.106 ± 0.038	0.009 ± 0.002	0.311 ± 0.073
CFQP (ours)	0.013 ± 0.001	$\mathbf{0.001 \pm 0.001}$	0.077 ± 0.050	$\mathbf{0.009 \pm 0.001}$	$\mathbf{0.002 \pm 0.001}$	$\mathbf{0.188 \pm 0.114}$
CFQP-GMM (ours)	$\mathbf{0.001 \pm 0.001}$	0.002 ± 0.001	$\mathbf{0.067 \pm 0.042}$	$\mathbf{0.009 \pm 0.001}$	$\mathbf{0.002 \pm 0.001}$	0.192 ± 0.0834

C Proof of Result 3.1

Our proof is structured as follows. We first show identifiability of non-parameteric mixtures at a single point $((X = x, T = t)$. We then show identifiability of the permutation function over $(\mathcal{X} \times \mathcal{T})$. Finally, based on these results, we derive the bound of Result 3.1.

C.1 Identifiability of Non-Parametric Mixture Models.

To show the identifiability in the non-additive case, we re-use notation and start from the outline derived in Aragam et al. [3].

C.1.1 Identifiability Result

We let (X, d) be a metric space and $(\mathcal{P}(X), \rho)$ the space of regular Borel probability measures on X with finite rth moments metrized by a metric ρ. Let's further define $\mathcal{P}^2(X) = \mathcal{P}(\mathcal{P}(X))$, the space of mixing measures over $\mathcal{P}(X)$. We represent a mixture distribution over X as a new probability measure $m(\cdot, \Lambda) \in \mathcal{P}(X)$ such that $m(A; \Lambda) = \int \gamma(A) d\Lambda(\gamma)$ for any $A \subset X$.

Given a Borel set $\mathfrak{L} \subset \mathcal{P}^2(X)$, we define $\mathcal{M}(\mathfrak{L}) := \{m(\Lambda) : \Lambda \in \mathfrak{L})$ as a family of mixture distributions over X. Because we are interested in finite mixtures, we use $\mathcal{P}_k^2 := \{\Lambda \in \mathcal{P}^2(X) : \|\text{supp}(\Lambda)\| \leq k\}$ to denote the set of mixing measures with k components. Finite mixtures $\Gamma \in \mathcal{M}(\mathcal{P}_k^2(X))$ can thus also be written in a more intuitive form, *i.e.* $\Gamma = \sum_{k=1}^{K} \omega_k \gamma_k$ where $\gamma_k \in \mathcal{P}(X)$ are the *mixture components* and ω_k are mixing weights, as in Equation 2 in the main text.

Note that we did not make any assumption regarding the distribution of the mixture components $(\gamma_k \in \mathcal{P}(X))$. Now, from [3], we have the following result.

Theorem C.1. *If $\mathfrak{L}$ is a $\mathfrak{Q}_L$-clusterable family, then there exists a function $h : \mathcal{M}(\mathfrak{L}) \to \mathfrak{L}$ such that $h(m(\Lambda)) = \Lambda$, where $m : \mathfrak{L} \to \mathcal{M}(\mathfrak{L})$ is the canonical embedding. In particular, m is a bijection and the mixture model $\mathcal{M}(\mathfrak{L})$ is identifiable.*

We see that identifiability of the mixture therefore hinges on the notion of *clusterability* of the individual distributions that compose the mixture. We now make the notion of clusterability more precise.

C.1.2 Clusterability

Projections Let's first introduce $\{\mathfrak{Q}_L\}_{L=1}^{\infty}$ as an indexed collection of families of mixing measures that satisfy the following requirements:

- $\mathfrak{Q}_L \subset \mathcal{P}_L^2(X)$ for each L;
- $\{\mathfrak{Q}_L\}$ is a filtration ($\mathfrak{Q}_L \subset \mathfrak{Q}_{L+1}$);
- The collection of mixture distributions $\mathcal{M}(\mathfrak{Q}_L)$ is identifiable for each L.

Examples of such a collection is the filtration of Gaussian mixtures with L components.

We then proceed by defining the ρ-projection of a mixture distribution $\Gamma = m(\Lambda)$ on a family of mixture distribution $\mathfrak{Q}_L$ by

$$T_L\Gamma = \{Q \in \mathfrak{Q}_L : \rho(Q, \Gamma) \leq \rho(P, \Gamma) \forall P \in \mathfrak{Q}_{\mathcal{L}}\}$$

where ρ is a metric on $\mathcal{P}(X)$ (*e.g.* the Hellinger distance). This projection intuitively maps a mixture distribution Γ to the closest element of the family $\mathfrak{Q}_L$ it is projected upon. We futher define the map $M_L : \mathfrak{Q}_{\mathcal{L}} \to \mathfrak{Q}_L$ that a projected distribution to its mixing measure:

$$M_L\Gamma = M_L(T_L\Gamma) = \{\Omega \in \mathfrak{Q}_L : m(\Omega) \in T_L\Gamma\}$$

Assignment Functions Because it is an element of $\mathfrak{Q}_L$, the projection $T_L\Gamma = Q^*$ is a mixture distribution and can thus be written as $T_L\Gamma = \sum_{l=1}^{L} \omega_l^* \gamma_l^*$. When projecting on mixture distributions with more mixture components than in the original distribution, *i.e.* $L \geq K$, one can then assign mixture components in Q^* to mixture components in Γ. We define the set of assignment maps $\alpha : [L] \to [K]$ as $\mathbb{A}_{L \to K}$. Given such an assignment map, we define $\psi_k(\alpha) := \sum_{l \in \alpha^{-1}(k)} \omega_l$ for the weights induced by the assignment map and $Q_k(\alpha) := \frac{1}{\psi_k(\alpha)} \sum_{l \in \alpha^{-1}(k)} \omega_l^* \gamma_l^*$ for the induced mixing components.

Regularity of a mixing measure We are now ready for introducing the definition of regularity of a mixing measure.

Definition C.1 (Regularity)**.** Suppose $\Lambda \in \mathcal{P}_K^2(X)$ and $\Gamma = m(\Lambda) \in \mathcal{M}_K(\mathcal{P}^2(X))$. The mixing measure Λ is called $\mathfrak{Q}_L$-regular if:

- (a) The ρ-projection $Q^* = T_L\Gamma$ exists and is unique for each L and $\lim_{L\to\infty} T_L\Gamma = \Gamma$;
- (b) There exists a sequence of assignment functions $\alpha = \alpha_L \in \mathbb{A}_{L \to K}$ such that $\lim_{L\to\infty} Q_k^*(\alpha) = \gamma_k \quad \text{and} \quad \lim_{L\to\infty} \psi_k^*(\alpha) = \omega_k \quad \forall k = 1, \ldots, K$

Any assignment function where (b) holds is called a *regular assignment*.

Clusterability We can now finally define clusterability of a family of mixing measures with respect to another regular family of mixing measures.

Definition C.2 (Clusterable family)**.** A family of mixing measures $\mathfrak{L} \subset \mathcal{P}^2(X)$ is called a $\mathfrak{Q}_L$-clusterable family, or just a clusterable family, if

- (a) Λ is $\mathfrak{Q}_L$-regular for all $\Lambda \in \mathfrak{L}$;

- (b) For all sufficiently large L, there exists a function $\chi_L : M_L(\mathfrak{L}) \to \mathbb{A}_{L\to K}$ such that $\chi_L(\Omega^*)$ is a regular assignment for every $\Lambda \in \mathfrak{L}$, with $\Omega^* = M_L(T_L\Gamma)$.

The resulting mixture model $\mathcal{M}(\mathfrak{L})$ is called a clusterable mixture model.

More intuitively, a family of mixing measures will then be $\mathfrak{Q}_L$-clusterable if one can project every mixing measure of the family onto $\mathfrak{Q}_L$ and if, for each of these projections, there exists a function that outputs a regular assignment for each mixture distribution given as input. This regular assignment is such that it clusters the elements of Q^* in a way that the distribution of each cluster converges to the distribution of a element of the original mixture distribution. In this case, one can *identify* a family of mixing measures by projecting each mixing distribution onto a regular family (*e.g.* Gaussian mixtures) and cluster it accordingly to χ_L. We refer the interested reader to [3] for more insight on clusterability and identifiability in non-parametric mixtures.

C.2 Proof of result 3.1

C.2.1 Identifiability at a Fixed Point

We start our proof by recalling the identifiability result in the case of clusterable families of mixing measures, whose proof can be found in [3].

Theorem C.2. *If $\mathfrak{L}$ is a $\mathfrak{Q}_L$-clusterable family, then there exists a function $h : \mathcal{M}(\mathfrak{L}) \to \mathfrak{L}$ such that $h(m(\Lambda)) = \Lambda$, where $m : \mathfrak{L} \to \mathcal{M}(\mathfrak{L})$ is the canonical embedding. In particular, m is a bijection and the mixture model $\mathcal{M}(\mathfrak{L})$ is identifiable.*

As laid out in the main text, we assume three observed random variables $X \in \mathcal{X}$, $Y \in \mathcal{Y}$ and $T \in \mathcal{T}$. As suggested by Assumption 1 in the main text, the conditional distribution of Y at each point $(X = x, T = t)$ is given by an unknown mixture distribution from a family $\mathcal{M}_K(\mathfrak{L})$ where each mixing measure $\Lambda \in \mathfrak{L}$ satisfies $\Lambda \in \mathcal{P}^2(Y)$ and $|\text{ supp}(\Lambda) |= K$. We write the distribution of Y conditioned on X and T as:

$$Y \mid X, T \sim \gamma(X, T) = \sum_{k=1}^{K} \omega_k \gamma_k(X, T) \quad \text{with } \gamma(X, T) \in \mathfrak{L} \tag{6}$$

Assuming that $\mathfrak{L}$ is clusterable, we can use Theorem C.2 to deduce that the mixture distribution $\gamma(X, T)$ is identifiable at each fixed point $(X = x, T = t)$.

Importantly, the idenfitifiablity insured by Theorem C.2 is up to a permutation of the mixture components. We write $\sigma_{x,t} : [K] \to [K]$ the permutation function at a specific point $(X = x, T = t)$. We define the assignment function that maps the elements of a mixture distribution $\gamma(X, T)$ to U_Z as $F_{(x,t)} : \mathcal{M}(\mathfrak{L}) \to \{[K], \mathcal{P}(Y)\}^K$. We now move to showing that the assignment function is identifiable up to a constant permutation $\bar{\sigma}$.

C.2.2 Identifiability of the Assignment Function

The support of random variables X and T, $\mathcal{X} \times \mathcal{T}$ is connected. Assuming further (Assumption 3) that $\mathfrak{L}$ is a clusterable family, we know that there exists a cluster function $\chi_{L,(x,t)} : \mathcal{M}_{\mathcal{L}}(\mathfrak{L}) \to \mathbb{A}_{L\to K}$ that maps components of the projected mixture onto the initial one. We can then define a combined operator that take a mixture distribution as input and returns an indexed sequence of mixture components.

Let $F_{(x,t)} : \mathcal{M}(\mathfrak{L}) \to \{[K], \mathcal{P}(Y)\}^K$ be this operator consisting of applying a ρ-projection onto a regular family, cluster the result according to $\chi_{L,(x,t)}$ and return the indexed estimated distributions $Q_k^*(\alpha)$. As the per the definition of regular mixing measures, we have also have $\lim_{L\to\infty} Q_k^*(\alpha) = \gamma_k$. Note that the indexing of the elements of the mixing distribution is arbitrary at a single point $(X = x, T = t)$ and thus defined up to a permutation $\bar{\sigma}$.

Because X and T are continuous on a connected domain, we can evaluate the operator at $(X = x + \delta x, T = t\delta_t)$.

$$F_{(x,t)} = \{(\sigma_{x,y}(i), q_i^*(x,t)) : i = 1, ..., K\}$$
$$F_{(x+\delta x, t+\delta t)} = \{(\sigma_{x+\delta x, y+\delta y}(i), q_i^*(x+\delta x, t+\delta t)) : i = 1, ..., K\}$$

What is more, for arbitrary small $(\delta x, \delta t)$ and $\forall (x,t) \in (\mathcal{X} \times \mathcal{T}); \forall k, k', k'' \in [K]$, we have

$$\rho(\gamma_k(x,t), \gamma_{k'}(x,t)) > \rho(\gamma_{k''}(x,t), \gamma_{k''}(x+\delta x, t+\delta t)$$

because we require the moments of each γ_k to be continuous in (x,t). It therefore implies a relation between $\sigma_{x,y}$ and $\sigma_{x+\delta x, y+\delta y}$ that satisfies:

$$\sigma_{x+\delta x, y+\delta y}(i) = \text{argmin}_j \rho(q_i^*(x,t), q_j^*(x+\delta x, t+\delta t))$$

The assignment function $F_{(x,t)}$ is thus identifiable up to a constant permutation $\bar{\sigma}$ that will determine all other permutations in $\mathcal{X} \times \mathcal{T}$.

C.2.3 Counterfactual Identifiability

To recapitulate, in the previous sections, we have shown (1) identifiability of a non-parametric mixture at a point $(X = x, T = t)$ based on the Assumption 3 and (2) identifiability of the permutation function over $(\mathcal{X} \times \mathcal{T})$. That is, for each point in $(X = x, T, t)$, we can identify the mixture components and the mixture weights and associate it to any other point $(X = x', T =')$, through the assignment function $F_{(x,t)}$.

In particular, we can identify the means μ_k of each cluster component γ_k. We have

$$\lim_{N\to\infty} \hat{\mu}_k(X,T) = \mathbb{E}_{Y\sim\gamma_k(X,T)}[Y] = \mu_k(X,T) \tag{7}$$

and

$$\lim_{N\to\infty} \hat{\omega}_k(X,T) = \omega_k(X,T). \tag{8}$$

Because the above estimators converge, the following posterior probability converges to the true value as well:

$$P(U_Z = u_z \mid Y = y, X = x, T = t) \tag{9}$$
$$= \frac{P(Y = y \mid U_Z = u_Z, X = x, T = t) P(U_Z = u_Z, \mid X = x, T = t)}{\sum_{k=1}^{K} P(Y = y \mid U_Z = k, X = x, T = t) P(U_Z = k, \mid X = x, T = t)} \tag{10}$$

for all $x \in \mathcal{X}$, $t \in \mathcal{T}$ and $y \in \mathcal{Y}$. Indeed, $P(Y = y \mid U_Z = u_Z, X = x, T = t)$ is identifiable, and $P(U_Z = u_Z, \mid X = x, T = t) = \omega_{u_Z}(x,t)$.

Let $\hat{\omega}_{u_z}(x,y,t)$ be the estimator of $P(U_Z = u_Z \mid Y = y, X = x, T = t)$ We are now ready to define the estimator for the counterfactual distribution $\nu_{t'}(x,y,t)$.

$$\nu_{t'}(x,y,t) = \sum_k \hat{\omega}_k(x,y,t) \delta(y = \hat{\mu}_k(x,t')). \tag{11}$$

This estimator is a discrete mixture distribution with mass on the means of the different mixture components, weighted by the posterior probability of the initial sample (x,t,y) belonging to a particular component.

C.2.4 Bounds

We can now use this estimator $\nu_{t'}(x, y, t)$ to derive the bounds of Result 3.1.

$$
\begin{aligned}
&\mathbb{E}_Y \left[W_1(\nu_{t'}(X,T,Y), Y'(X,T,Y)) \right] \\
&\leq \mathbb{E}_Y \left[\iint_{y', \nu_{t'}} \|y' - \nu_{t'}(X,T,Y)\|_2 P(Y' = y', \nu_{t'} \mid X,T,Y)\, dy'\, d\nu_{t'} \right] \\
&= \int_y \left(\iint_{y', \nu_{t'}} \|y' - \nu_{t'}(X,T,Y)\|_2 p(Y' = y', \nu_{t'} \mid X,T,Y)\, dy'\, d\nu_{t'} \right) P(Y = y \mid X,T)\, dy \\
&= \int_y \left(\iint_{y', \nu_{t'}} \|y' - \nu_{t'}(X,T,Y)\|_2 P(Y' = y' \mid X,T,Y) p(\nu_{t'} \mid X,T,Y)\, dy'\, d\nu_{t'} \right) P(Y = y \mid X,T)\, dy \qquad (12) \\
&= \iint_{y', \nu_{t'}} \|y' - \nu_{t'}(X,T,Y)\|_2 p(Y' = y' \mid X,T) P(\nu_{t'} \mid X,T,Y) dy' d\nu_{t'} \\
&= \sum_{u_Z=1}^{K} P(U_Z = u_Z \mid X,T,Y) \cdot \\
&\qquad \iint_{y', \nu_{t'}} \|y' - \nu_{t'}(X,T,Y)\|_2 p(Y' = y' \mid X,T,U_Z = u_Z) p(\nu_{t'} \mid X,T,U_Z = u_Z)\, dy'\, d\nu_{t'} \qquad (13)
\end{aligned}
$$

Step 12 is obtained by conditional independence between y' and $\nu_{t'}$ conditioned on X, T, Y. Step 13 is obtained by law of total probabilities. Step 13 is obtained by factorizing on U_Z. Note that $P(Y' = y' \mid X, T, U_z = u_z)$ corresponds to the mixture component γ_k and the posterior assignment $P(U_Z = u_Z \mid X, T, Y)$ corresponds to $\hat{\omega}_{u_Z}(X, Y, T)$, as per the definition above.

What is more, conditioned on U_Z, $\nu_{t'}(X, T, U_z = k)$ reduces to a point estimate at $\hat{\mu}_k(X, T)$. We can then write

$$
\begin{aligned}
&\mathbb{E}_Y \left[W_1(\hat{\nu_{t'}}(X,T,Y), Y'(X,T,Y)) \right] \\
&\leq \sum_{u_Z=1}^{K} \hat{\omega}_{u_Z}(X,Y,T) \int_{y'} \int_{\nu_{t'}} \| \, y' - \nu_{t'}(X,T,Y) \, \| \, p(Y' = y' \mid X,T,U_Z = u_Z) p(\nu_{t'} \mid X,T,U_Z = u_Z) dy' d\nu_{t'} \\
&= \sum_{u_Z=1}^{K} \hat{\omega}_{u_Z}(X,Y,T) \int_{y'} \int_{\nu_{t'}} \| \, y' - \nu_{t'}(X,T,Y) \, \| \, \gamma_k(y' \mid X,T,U_Z = u_Z) p(\nu_{t'} \mid X,T,U_Z = u_Z) dy' d\nu_{t'} \\
&= \sum_{u_Z=1}^{K} \hat{\omega}_{u_Z}(X,Y,T) \int_{y'} \int_{\nu_{t'}} \| \, y' - \nu_{t'}(X,T,Y) \, \| \, \gamma_k(y' \mid X,T,U_Z = u_Z) \delta(\nu_{t'} = \hat{\mu}_k(X, T = t')) dy' d\nu_{t'} \\
&= \sum_{u_Z=1}^{K} \hat{\omega}_{u_Z}(X,Y,T) \int_{y'} \| \, y' - \hat{\mu}_k(X, T = t') \, \| \, \gamma_k(y' \mid X,T,U_Z = u_Z) dy' \\
&= \sum_{u_Z=1}^{K} \hat{\omega}_{u_Z}(X,Y,T) W_1^k(\hat{\mu}_k, \gamma_k) \qquad (14) \\
&\leq \max_k W_1^k(\hat{\mu}_k, \gamma_k) \qquad (15) \\
&= \delta
\end{aligned}
$$

Where 15 holds because it 14 is a convex combination $W_1^k(\hat{\mu}_k, \gamma_k)$. This ends the proof of Result 3.1.

C.2.5 The additive noise case

Result 3.1 mentions the bounds reduces to zero $\delta = 0$ when the noise is additive. From identifiability at of a mixture at a single point, we had:

$$\lim_{N\to\infty} \hat{\mu_k}(X,T) = \mathbb{E}_{Y\sim\gamma_k(X,T)}[Y] = \mu_k(X,T), \tag{16}$$

$$\lim_{N\to\infty} \hat{\omega}_k(X,T,Y) = P(U_Z = u_z \mid X,T,Y). \tag{17}$$

We also have convergence of the covariance matrix estimator

$$\lim_{N\to\infty} \hat{\Sigma_k}(X,T) = \mathbb{V}\text{ar}_{Y\sim\gamma_k(X,T)}[Y] = \Sigma_k(X,T). \tag{18}$$

In the additive noise setup, the distribution of Y being normally distributed, the value of U_η is actually identifiable. Let us recall the definition of the counterfactual distribution (Eq. 1) in the main text.

$$P(Y_{t'} = y' \mid X = x, Y = y, T = t) = \int_u P(Y_{t'}(u) = y')P(U = u \mid X = x, T = t, Y = y)$$

In the additive case, we have $Y \mid X = x, T = t, U_Z = u_Z = \Sigma_{u_Z}(x,t)U_\eta + \mu_k(x,t)$ with $U_\eta \sim \mathcal{N}(0, \mathbf{1})$. The counterfactual density can thus be written as

$$\nu_{t'}(x,t,y) = \sum_k P(U_Z = k \mid x,y,t) \cdot \delta(u_{\eta,k} = \Sigma_k(x,t)^{-1}y - \mu_k(x,t)) \cdot \delta(y' = \Sigma_k(x,t')u_{\eta,k} + \mu_k(x,t')) \tag{19}$$

Considering the following estimator for the counterfactual distribution:

$$\hat{\nu}_{t'}(x,t,y) = \sum_k \hat{\omega}_k(x,t,y) \cdot \delta(u_{\eta,k} = \hat{\Sigma}_k(x,t)^{-1}y - \hat{\mu}_k(x,t)) \cdot \delta(y' = \hat{\Sigma}_k(x,t')u_{\eta,k} + \hat{\mu}_k(x,t'))$$

We can use Equations 16, 17 and 18 to show that our estimator converges to the true distribution in the limit of infinite number of samples:

$$\lim_{N\to\infty} \hat{\nu}_{t'}(x,t,y) = \nu_{t'}(x,t,y) \tag{20}$$

Therefore, the distance between the estimated and true counterfactual distributions reduces to 0:

$$\lim_{N\to\infty} W_1(\hat{\nu}_{t'}(x,t,y), \nu_{t'}(x,t,y)) = 0. \tag{21}$$

This concludes our proof of Result 3.1.