GAUSSIAN MIXTURE COUNTERFACTUAL GENERATOR

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating synthetic control arms is a key challenge in clinical research, particularly in crossover trials where placebo data becomes unavailable after patients switch to active treatment. The absence of placebo data complicates estimating long-term efficacy and safety. To solve this, we propose a Gaussian mixture model that generates counterfactual data without needing control data for training. This method handles time-varying, continuous doses and estimates effects between treatment switchers and an extended placebo group, providing valuable insights for treatment effects, evidence generation, and decision-making.

1 INTRODUCTION

A crossover trial is a valuable design in clinical research, particularly for evaluating the long-term efficacy and safety of a new medication. Initially, a randomized clinical trial (RCT) is conducted in which participants are randomly assigned to either the active treatment group or the placebo group. After the initial phase, participants in the placebo group switch to the active treatment arm for additional weeks or months in an open-label crossover trial. This design allows researchers to collect more comprehensive data on the effects of the medication and provides ethical benefits by eventually offering active treatment to all participants (Elbourne et al., 2002; Nolan et al., 2016; Dwan et al., 2019).

The need for such extended trials arises from the desire to understand the sustained impact of a treatment beyond the initial trial period. By including a crossover phase, researchers can observe the effects of the medication on participants who were initially on placebo, thereby enhancing the robustness of the findings. This approach can reveal whether the benefits observed in the initial RCT phase are consistent over a longer period and help identify any delayed adverse effects. In addition, it allows for a more ethical study design, as all participants eventually receive active treatment, which is particularly important in studies involving serious or chronic conditions.

However, there are still challenges associated with crossover trials. An issue is the potential for carryover effects, where the impact of the initial treatment phase influences the outcomes in the crossover phase. This can complicate the interpretation of results. Another problem is the increased complexity and duration of the trial, which can lead to higher costs and logistical challenges. Ensuring participant adherence over a longer period can also be difficult, which can affect the reliability of the data (Mills et al., 2009; Heeson, 2020).

One significant challenge associated with crossover trials is the difficulty in accurately estimating the effects of medications during the extended phase for patients who switched from placebo to active treatment (Zhou et al., 2024). This arises because once the crossover occurs, there is no longer a placebo group to serve as a control for comparison. Without ongoing placebo data, it becomes challenging to distinguish between the effects of the medication and other factors such as natural disease progression or placebo effects that might have persisted. This lack of a concurrent control group can complicate the interpretation of the results of the extended phase, which can lead to less robust conclusions about the long-term efficacy and safety of the medication. Consequently, researchers must carefully consider these limitations when designing and analyzing extended crossover trials.

To address the challenge of estimating the effects between switched patients and a potential extended placebo group, researchers can utilize several strategies. An approach is to use external placebo data from historical trials, which can serve as a benchmark for comparison (Letailleur et al., 2023). This method leverages existing data to create a virtual placebo group, allowing a more accurate estimation of treatment effects during the extended phase (Serrano et al., 2023). However, this

solution may not always be feasible, especially for rare diseases where historical data is scarce or nonexistent. In such cases, researchers might consider adaptive trial designs (Pallmann et al., 2018) that allow modifications based on interim results or Bayesian statistical methods (Berry et al., 2010) that incorporate prior knowledge and expert opinion to enhance the analysis. Despite these potential solutions, challenges remain, including ensuring the comparability of external data and addressing biases that may arise from differences in study populations or methodologies. Therefore, careful planning and robust statistical techniques are essential to mitigate these issues and derive meaningful conclusions from extended crossover trials.

The issues described in the crossover trials highlight the broader types and complexity of the design of synthetic control arms in clinical studies. In modern trial settings, generating synthetic data remains a challenging problem that machine learning models have yet to fully address. This paper proposes a model capable of generating synthetic control data by *training solely on active arm data*. The approach seeks to overcome limitations in existing methods that require both control and active data. This model aims to provide a robust alternative for trial designs lacking a conventional control group using advanced learning techniques.

2 Related Work

2.1 Limitation of SCM in Creating a Synthetic Control Arm

Generating synthetic control arms in crossover trials is challenging due to the unavailability of control group data after patients switch to active treatment. In a standard RCT, control group data are crucial for comparing outcomes over time. However, in crossover trials—especially those with an open-label extension phase—patients initially assigned to the control group often switch to the active treatment after a certain period (e.g., after 12 weeks). This switch means that there are no remaining patients in the control group during the extended phase (e.g., up to 24 weeks). The absence of extended control data makes it difficult to assess the long-term efficacy and safety of the treatment, as there is no direct comparison group.

Abadie's synthetic control method (SCM) attempts to approximate the treatment group's outcomes by creating a weighted combination of control units that closely resembles the treatment-assigned patient data before the intervention (Abadie & Gardeazabal, 2003; Abadie et al., 2010). This method relies on the availability of control data to compute the necessary weights. Applying Abadie's method becomes problematic in the context of crossover trials lacking extended control data. For example, in a trial with a 12-week RCT followed by a 12-week open-label extension where all patients receive the active treatment, no 24-week control data is available. Without this data, we cannot calculate the weighted sum of the 24-week control outcomes needed to construct a synthetic control arm for the switched patients. This limitation prevents the direct application of Abadie's algorithm in such scenarios. In Zhou et al. (2024), SCM and difference-in-differences (Abadie, 2005; Sant'Anna & Zhao, 2020) methods successfully measured treatment effects in crossover trials with the availability of external controls; however, this approach is not feasible in studies where securing external controls is challenging.

2.2 S-Learner vs. T-Learner

In common types of meta-learner, two distinct concepts are prevalent: the S-learner and the T-learner (Künzel et al., 2019; Okasa, 2022). The term 'S-learner' is derived from the notion of a single base learner, while 'T-learner' is an abbreviation for two base learners (Shalit et al., 2017). In a straightforward scenario involving a treatment group and a control group within a patient data set, the T-learner estimates two separate base learners: one for the treatment group data and another for the control group data. Subsequently, it computes the difference between these two base learners. Estimating individual treatment effects (ITE) based on a Gaussian mixture model exemplifies a T-learner in the context of counterfactual generation (Ahn & Vashist, 2024). The GMM-based counterfactual generator demonstrated superior performance compared to the synthetic control method on simulated low-density lipoprotein (LDL) data (Qian et al., 2021) and showed robustness even when dealing with heavily biased datasets.

However, the algorithm fundamentally operates as a treatment-controlled method, requiring a learned data distribution model on control units to generate counterfactual outcomes. This reliance

poses limitations in scenarios where control data are unavailable, such as in crossover trials with no extended control group. Additionally, its application as a T-learner is confined to data sets with a small, finite number of groups. As the number of groups increases, preparing distinct base learners, each trained on different group data, becomes necessary. A significant limitation of such a T-learner is its inability to accommodate more complex real-world data, where potential treatments can span a multivariate, time-varying, and continuous domain.

Despite this limitation, the algorithm provides an important conceptual foundation for our work. Specifically, it functions as a T-learner, assuming separate distribution models for treatment and control data. Adapting the model into an S-learner framework allows us to develop a unified model capable of counterfactual inference even for unobserved treatment types. This adaptation allows us to generate counterfactual data without relying on control data for training. Therefore, our proposed method builds upon the foundational idea, extending its applicability to control data-unavailable situations.

3 MODEL

In this section, we outline a latent variable model known as static state analysis (SSA), originally used in the paper by (Ahn & Vashist, 2024), and present how our new ideas are incorporated into this model. Building on the methodologies, we introduce two key problems. The first problem involves a brief overview of how to train the SSA model. The second problem concerns what counterfactual predictions to generate using the trained model.

3.1 STATIC STATE ANALYSIS (SSA)

The static state analysis decomposes a longitudinal dataset $\{\boldsymbol{x}^{(t)}\}$ into a time-varying observer matrix $\boldsymbol{W}^{(t)}$ and time-independent state vectors $\boldsymbol{s}$ with additive noises $\boldsymbol{\eta}^{(t)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}^{(t)})$:

$$\boldsymbol{x}^{(t)} = \boldsymbol{W}^{(t)}\boldsymbol{s} + \boldsymbol{\eta}^{(t)}. \tag{1}$$

The time-independent state $\boldsymbol{s}$ is referred to as a static state. In this paper, longitudinal data $\boldsymbol{x}^{(t)}$ generally concatenate all different types of observed data, including baseline data $\boldsymbol{v}$ (e.g., demographic and genetic information), time-dependent covariates $\boldsymbol{l}^{(t)}$, results $\boldsymbol{y}^{(t)}$, and treatments $\boldsymbol{a}^{(t)}$. $\boldsymbol{x}^{(t)}$ and $\boldsymbol{W}^{(t)}$ can be written as $\boldsymbol{x}^{(t)} = [\boldsymbol{a}^{(t)};\ \boldsymbol{l}^{(t)};\ \boldsymbol{y}^{(t)}]$ and $\boldsymbol{W}^{(t)} = [\boldsymbol{W}_a^{(t)};\ \boldsymbol{W}_l^{(t)};\ \boldsymbol{W}_y^{(t)}]$ for $t \geq 1$. Equation (1) is decomposed into

$$\boldsymbol{a}^{(t)} = \boldsymbol{W}_a^{(t)}\boldsymbol{s} + \boldsymbol{\eta}_a^{(t)}, \tag{2}$$

$\boldsymbol{l}^{(t)} = \boldsymbol{W}_l^{(t)}\boldsymbol{s} + \boldsymbol{\eta}_l^{(t)}$, and $\boldsymbol{y}^{(t)} = \boldsymbol{W}_y^{(t)}\boldsymbol{s} + \boldsymbol{\eta}_y^{(t)}$ with $\boldsymbol{x}^{(0)} \equiv \boldsymbol{v} = \boldsymbol{W}_v^{(t)}\boldsymbol{s} + \boldsymbol{\eta}_v^{(t)}$. Equation (2) is our new equation that is directly included as observational data, whereas the original SSA model did not include it. Although it may appear to be a mere addition of a simple formula, it is important to note that this single addition fundamentally alters the nature and scope of the algorithm. Equations (1) and (2) also represent a factor model where the noise model follows a diagonal noise covariance matrix. This paper is not the first to model treatments or causes directly using a factor model. Wang & Blei (2019) and Bica et al. (2020a) successfully removed hidden confounding factors by directly modeling multiple causes as latent variables.

3.2 PROBLEM I: FITTING OF A MODEL TO A DATA SET

Given an observed data set $\mathbb{D} = \{\boldsymbol{x}_n^{(t)} | t \in \mathbb{T}_n, n = 1, \cdots, N\} \subset \mathcal{D}$, our first problem is to perform SSA by decomposing the data set $\mathbb{D}$ into a time-varying matrix $\boldsymbol{W}^{(t)}$ and a set of static state vectors $\mathbb{S} = \{\boldsymbol{s}_n | n = 1, \cdots, N\} \subset \mathcal{S}$. To learn the static state space $\mathcal{S}$ from the dataset, we employ a Gaussian mixture model (GMM) probabilistic distribution to model the static state as a vector-valued random variable, which is given by

$$p(\mathbf{s}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{3}$$

There is no more dependence on treatment notation because treatment data are directly included in the observational data. Now, the problem is to estimate a set of parameters $\mathbb{Q} =$

$\{\boldsymbol{W}, \pi_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \boldsymbol{\Psi}\}$ that maximizes the expected complete-data log-likelihood by using the expectation and maximization algorithm:

$$\hat{\mathbb{Q}} = \arg\max_{\mathbb{Q}} \sum_{n=1}^{N} \sum_{t \in \mathbb{T}_n} \langle \ln p(\boldsymbol{x}_n^{(t)}, \mathbf{s}_n | \mathbb{Q}) \rangle \tag{4}$$

where $\langle \mathbf{s}_n \rangle = \int |d\mathbf{s}| p(\mathbf{s}|\boldsymbol{x}_n, \mathbb{Q}) \times \mathbf{s}$ and $\langle \mathbf{s}_n \mathbf{s}_n^T \rangle = \int |d\mathbf{s}| p(\mathbf{s}|\boldsymbol{x}_n, \mathbb{Q}) \times \mathbf{s}\mathbf{s}^T$. Then, we can also get the noise vector as $\langle \boldsymbol{\eta}_n^{(t)} \rangle = \boldsymbol{x}_n^{(t)} - \widehat{\boldsymbol{W}}^{(t)} \langle \mathbf{s}_n \rangle$.

3.3 Problem II: Counterfactual Generation with No Backward Causation

Let $\hat{\mathbb{Q}}$ be a maximizer of the log-likelihood function by Eq. (4) for a factual data set $\mathbb{D}_{\text{f}} = \{\boldsymbol{x}_{n,\text{f}}^{(:)} | n = 1, \cdots, N\}$. The main problem in our paper is to generate a synthetic counterfactual longitudinal dataset $\mathbb{D}_{\text{cf}}^{(\tau+\delta:)} = \{\boldsymbol{x}_{n,\text{cf}}^{(\tau+\delta:)} | n = 1, \cdots, N\}$ from using $\hat{\mathbb{Q}}$ when we assume that we had assigned alternative treatments $\boldsymbol{a}_{n,\text{cf}}^{(\tau:\tau+\delta)}$ instead of $\boldsymbol{a}_{n,\text{f}}^{(\tau:\tau+\delta)}$ where we define the colon notation as concatenating time-varying data or parameters:

$$\boldsymbol{x} \equiv \boldsymbol{x}^{(:)} \equiv \begin{bmatrix} \boldsymbol{x}^{(0)} \\ \boldsymbol{x}^{(1)} \\ \vdots \end{bmatrix}, \quad \boldsymbol{x}^{(:\tau)} \equiv \boldsymbol{x}^{(0:\tau)} \equiv \begin{bmatrix} \boldsymbol{x}^{(0)} \\ \vdots \\ \boldsymbol{x}^{(\tau-1)} \end{bmatrix}, \quad \boldsymbol{a}^{(\tau:\tau+\delta)} \equiv \begin{bmatrix} \boldsymbol{a}^{(\tau)} \\ \vdots \\ \boldsymbol{a}^{(\tau+\delta-1)} \end{bmatrix}. \tag{5}$$

Counterfactual pretreatment data must align as closely as possible with the provided factual pretreatment data (Abadie & Gardeazabal, 2003; Doudchenko et al., 2021). Thus, equivalently, the above problem can be written as generating an entire dataset $\mathbb{D}_{\text{cf}} = \{\boldsymbol{x}_{n,\text{cf}}^{(:)} | n = 1, \cdots, N\}$ that satisfies $\boldsymbol{x}_{n,\text{cf}}^{(:\tau)} = \boldsymbol{x}_{n,\text{f}}^{(:\tau)}$.

The subscript "f" stands for factual, and "cf" is derived from the first letters of counter-factual. Data obtained through observation is called factual data. On the other hand, data generated through counterfactual thinking without being observed in the real world is referred to as counterfactual data in this paper. Counterfactual thinking involves considering hypothetical scenarios, such as imagining what would happen if a patient who took medication at time τ had not taken it. Before time τ, the patient actually did not take the medication, so the factual data and counterfactual data are the same, which can be expressed as $\boldsymbol{x}_{n,\text{cf}}^{(:\tau)} = \boldsymbol{x}_{n,\text{f}}^{(:\tau)}$. The goal of this paper is to obtain counterfactual data. This can be interpreted in the context of clinical trials as virtual patient data or a synthetic placebo arm. In the next section, we introduce the basic assumptions of counterfactual thinking and propose a new method for counterfactual prediction.

4 Methods

To solve the second problem, this section details the specific assumptions and formulas for generating counterfactual predictions.

4.1 Assumptions

A probabilistic observation refers to the process of observing outcomes that are not deterministic but rather governed by probability distributions. In other words, instead of a single, fixed outcome, there are multiple possible outcomes, each with a certain probability of occurring. This concept is fundamental in fields like statistics, machine learning, and various branches of science and engineering (Gelman et al., 1995). The process of obtaining data through observation can be also described using the following interpretation of the SSA model.

Probabilistic Observation

Consider a patient with factual data $\boldsymbol{x}_{\mathrm{f}}^{(:\tau)}$ has received a course of treatment $\boldsymbol{a}_{\mathrm{f}}^{(\tau:\tau+\delta)}$. Post-treatment factual data can be obtained by sampling the patient's state $\mathbf{s}_{\mathrm{f}} \sim p(\mathbf{s}|\boldsymbol{x}_{\mathrm{f}}^{(:\tau)}, \boldsymbol{a}_{\mathrm{f}}^{(\tau:\tau+\delta)})$ at the realization $\boldsymbol{s}_{\mathrm{f}}$:

$$\boldsymbol{x}_{\mathrm{f}}^{(\tau:)} = \boldsymbol{W}^{(\tau:)}\boldsymbol{s}_{\mathrm{f}} + \boldsymbol{\eta}^{(\tau:)} \tag{6}$$

where $\boldsymbol{\eta}^{(\tau:)}$ is a measurement noise vector.

This statement implies a probabilistic observation process in which future outcomes are probabilistically determined by a series of treatments given at the current time τ. However, this statement only addresses the probabilistic acquisition of factual data and requires separate counterfactual thinking to handle counterfactual data. It is as follows:

Retrospective Counterfactual Thinking

Suppose the patient with factual data $\boldsymbol{x}_{\mathrm{f}}$ had instead received an alternative treatment $\boldsymbol{a}_{\mathrm{cf}}^{(\tau:\tau+\delta)} \neq \boldsymbol{a}_{\mathrm{f}}^{(\tau:\tau+\delta)}$. We would then have obtained counterfactual data through the same observational process at the realization $\boldsymbol{s}_{\mathrm{cf}}$:

$$\boldsymbol{x}_{\mathrm{cf}}^{(\tau:)} = \boldsymbol{W}^{(\tau:)}\boldsymbol{s}_{\mathrm{cf}} + \boldsymbol{\eta}^{(\tau:)} \tag{7}$$

$$\boldsymbol{s}_{\mathrm{cf}} \sim p(\mathbf{s}|\boldsymbol{x}_{\mathrm{f}}, \boldsymbol{a}_{\mathrm{cf}}^{(\tau:\tau+\delta)}). \tag{8}$$

Here, $\boldsymbol{\eta}^{(\tau:)}$ represents the same noise term as in the factual data observation from Eq. (6), reflecting the assumption that the same observational and measurement processes apply, albeit under the alternative treatment scenario.

Retrospective counterfactual thinking basically refers to the mental process of imagining alternative outcomes to past events, essentially considering "what might have been" if different actions or circumstances had occurred. The current point in time at which counterfactual thinking is attempted is in the future relative to time τ. In other words, τ is a point in the past. Equations (7) and (8) generate counterfactual data for times after τ, assuming that the counterfactual data before τ are exactly the same as the factual data observed before τ.

4.2 Generating a Counterfactual State

Once we find a maximizer $\hat{\mathbb{Q}}$ estimated by solving problem I, we can sample a factual state $\boldsymbol{s}_{\mathrm{f}}$ from $p(\mathbf{s}|\boldsymbol{x}_{\mathrm{f}}^{(:\tau)})$ or obtain its expectation $\boldsymbol{s}_{\mathrm{f}} \equiv \langle \mathbf{s}_{\mathrm{f}} \rangle$ for a patient with data $\boldsymbol{x}_{\mathrm{f}}^{(:\tau)}$. As the first step in counterfactual data generation for the patient, we can generate a counterfactual state by realizing a vector $\boldsymbol{s}_{\mathrm{cf}}$ from Eq. (8)

$$\mathbf{s}_{\mathrm{cf}} \sim \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{s}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{9}$$

where $\mathbf{s}_{\mathrm{cf}}$ is conditioned on

$$\begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}_{\mathrm{cf}}^{(\tau:\tau+\delta)} - \boldsymbol{a}_{\mathrm{f}}^{(\tau:\tau+\delta)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{W}^{(:\tau)} \\ \boldsymbol{W}_a^{(\tau:\tau+\delta)} \end{bmatrix} (\mathbf{s}_{\mathrm{cf}} - \boldsymbol{s}_{\mathrm{f}}). \tag{10}$$

Equivalently, we can write a counterfactual state random vector in the form of

$$\begin{aligned} \mathbf{s}_{\mathrm{cf}} &= \boldsymbol{s}_{\mathrm{f}} + \begin{bmatrix} \boldsymbol{W}^{(:\tau)} \\ \boldsymbol{W}_a^{(\tau:\tau+\delta)} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{a}_{\mathrm{cf}}^{(\tau:\tau+\delta)} - \boldsymbol{a}_{\mathrm{f}}^{(\tau:\tau+\delta)} \end{bmatrix} + \ker\left(\begin{bmatrix} \boldsymbol{W}^{(:\tau)} \\ \boldsymbol{W}_a^{(\tau:\tau+\delta)} \end{bmatrix} \right) \boldsymbol{\xi} \\ &\equiv \boldsymbol{s}_{\mathrm{f}} + \boldsymbol{\delta}_a + \boldsymbol{N}\boldsymbol{\xi} \end{aligned} \tag{11}$$

where dagger and ker() denote the pseudo-inverse and a basis of the kernel of $[\boldsymbol{W}^{(:\tau)};\ \boldsymbol{W}_a^{(\tau:\tau+\delta)}]$, respectively, and $\boldsymbol{\xi}$ is a random vector of the following distribution by denoting the treatment difference on the pseudo-inverse basis as $\boldsymbol{\delta}_a$ and the kernel matrix as $\boldsymbol{N}$:

$$\boldsymbol{\xi} \sim \sum_{k=1}^{K} p_k \mathcal{N}(\boldsymbol{\xi}; \boldsymbol{m}_k, \boldsymbol{C}_k) \tag{12}$$

with

$$\boldsymbol{C}_k = (\boldsymbol{N}^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{N})^{-1} \tag{13}$$

$$\boldsymbol{m}_k = \boldsymbol{C}_k \boldsymbol{N}^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu}_k - \boldsymbol{s}_\mathrm{f} - \boldsymbol{\delta}_a) \tag{14}$$

$$p_k = \frac{\pi_k \mathcal{N}(\boldsymbol{N}\boldsymbol{m}_k + \boldsymbol{s}_\mathrm{f} + \boldsymbol{\delta}_a | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) |\boldsymbol{C}_k|^{\frac{1}{2}}}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\boldsymbol{N}\boldsymbol{m}_{k'} + \boldsymbol{s}_\mathrm{f} + \boldsymbol{\delta}_a | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}) |\boldsymbol{C}_{k'}|^{\frac{1}{2}}}. \tag{15}$$

Notice that $\boldsymbol{\delta}_a$ in Equations (11), (14), and (15) is the new term derived by introducing Equation (2), and makes a crucial contribution to this paper.

4.3 Generating Counterfactual Outcomes

From Equation (6), (7), and (11), we can generate a counterfactual data vector based on the Gaussian mixture: $\boldsymbol{x}_\mathrm{cf} = \boldsymbol{x}_\mathrm{f} + \boldsymbol{W}(\boldsymbol{\delta}_a + \boldsymbol{N}\boldsymbol{\xi})$ where $\boldsymbol{\xi}$ is a realization of Equation (12). More formally, we can state:

Gaussian Mixture Counterfactual Generator

If a patient with factual data $\boldsymbol{x}_\mathrm{f}$ had received a treatment differing by $\boldsymbol{\delta}_a$, the counterfactual data that we would have obtained is represented as a random variable given by

$$\mathbf{x}_\mathrm{cf} \sim \sum_{k=1}^{K} p_k \mathcal{N}\left(\mathbf{x}; \boldsymbol{x}_\mathrm{f} + \boldsymbol{W}(\boldsymbol{\delta}_a + \boldsymbol{N}\boldsymbol{m}_k), \boldsymbol{W}\boldsymbol{N}\boldsymbol{C}_k\boldsymbol{N}^T\boldsymbol{W}^T\right) \tag{16}$$

This corresponds to the *Gaussian Mixture Counterfactual Generator* (GMCG), defined with the parameters $\hat{\mathbb{Q}}$ from Equation (4).

This formulation reflects a probabilistic approach for generating counterfactual outcomes based on a *single* Gaussian mixture model. Note that the GMCG presented in Eq. (16) is not a statically fixed GMM model but a model where $\boldsymbol{\delta}_a$, $\boldsymbol{N}$, $\boldsymbol{C}_k$, and $\boldsymbol{m}_k$ dynamically change according to the given treatment $\boldsymbol{a}_\mathrm{cf}^{(\tau:\tau+\delta)}$, intervention time τ, interval δ, etc.

Because $\boldsymbol{W}^{(:\tau)}\boldsymbol{\delta}_a = \boldsymbol{W}^{(:\tau)}\boldsymbol{N} = \mathbf{0}$, the random variable $\mathbf{x}_\mathrm{cf}^{(:\tau)}$ is always the same as $\boldsymbol{x}_\mathrm{f}^{(:\tau)}$ by Equation (16). However, the difference in outcomes made by a particular vector $\boldsymbol{\xi}$ in the probabilistic model in Equation (12) is a time-varying *individualized treatment effect* (ITE) caused by the difference in treatments $\boldsymbol{a}_\mathrm{cf}^{(\tau:\tau+\delta)} - \boldsymbol{a}_\mathrm{f}^{(\tau:\tau+\delta)}$ that we can obtain

$$\mathrm{ITE} \equiv \boldsymbol{y}_\mathrm{cf}^{(\tau:)} - \boldsymbol{y}_\mathrm{f}^{(\tau:)} = \boldsymbol{W}_y^{(\tau:)}(\boldsymbol{\delta}_a + \boldsymbol{N}\boldsymbol{\xi}). \tag{17}$$

The expected ITE (eITE) we can obtain on average is

$$\mathbb{E}[\mathrm{ITE}] \equiv \mathbb{E}[\mathbf{y}^{(\tau:)} | \boldsymbol{x}_\mathrm{f}, \boldsymbol{a}_\mathrm{cf}^{(\tau:\tau+\delta)}] - \boldsymbol{y}_\mathrm{f}^{(\tau:)} = \boldsymbol{W}_y^{(\tau:)}(\boldsymbol{\delta}_a + \boldsymbol{N}\sum_{k=1}^{K} p_k \boldsymbol{m}_k). \tag{18}$$

By using Equations (12), (14), and (15), we can calculate the *conditional averaged treatment effect* (CATE) from Eq. (17)

$$\begin{aligned} \mathrm{CATE} &\equiv \mathbb{E}[\mathbf{y}^{(\tau+\delta:)} | \boldsymbol{x}_\mathrm{f}^{(:\tau)}, \boldsymbol{a}_\mathrm{cf}^{(\tau:\tau+\delta)}] - \mathbb{E}[\mathbf{y}^{(\tau+\delta:)} | \boldsymbol{x}_\mathrm{f}^{(:\tau)}, \boldsymbol{a}_\mathrm{f}^{(\tau:\tau+\delta)}] \\ &= \boldsymbol{W}_y^{(\tau:)}\boldsymbol{N}\sum_{k=1}^{K}(p_k\boldsymbol{m}_k - q_k\boldsymbol{n}_k) \end{aligned} \tag{19}$$

where

$$\boldsymbol{n}_k = \boldsymbol{C}_k \boldsymbol{N}^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{\mu}_k - \boldsymbol{s}_\mathrm{f}) \tag{20}$$

$$q_k = \frac{\pi_k \mathcal{N}(\boldsymbol{N}\boldsymbol{n}_k + \boldsymbol{s}_\mathrm{f} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) |\boldsymbol{C}_k|^{\frac{1}{2}}}{\sum_{k'=1}^{K} \pi_{k'} \mathcal{N}(\boldsymbol{N}\boldsymbol{n}_{k'} + \boldsymbol{s}_\mathrm{f} | \boldsymbol{\mu}_{k'}, \boldsymbol{\Sigma}_{k'}) |\boldsymbol{C}_{k'}|^{\frac{1}{2}}}. \tag{21}$$

5 EXPERIMENTS

5.1 ILLUSTRATIVE EXAMPLE OF A CROSSOVER TRIAL

5.1.1 LINEAR EFFICACY

This section demonstrates how the proposed GMCG algorithm by Equation (16) can be applied to crossover trial data to generate synthetic control data for illustrative purposes. The data set shown here is not actual but derived from a simulation model, the details of which can be found in Appendix C. In the left panel of Figure 1, data from two arms are depicted using shades of red and blue. The red-shaded arm represents groups that received the assigned doses over an entire 24-week period, while the blue-shaded arm represents groups that received the doses only during the last 12 weeks. Each arm is further subdivided into groups based on different doses of 1 mg and 2 mg, resulting in a total of four distinct dose groups. Notably, no group remains on placebo for the entire duration, necessitating the creation of a synthetic control arm to compare to the treatment arms to estimate the efficacy of the treatment. The GMCG model was trained on 400 patients' outcome data across the four groups, each consisting of 100 patients, without additional covariates.

The four thick lines of different colors on the left are randomly selected and displayed, and they are further dealt with in the four subplots on the right. The first subplot displays data from the 1mg group during the second 12-week period, with the algorithm-generated synthetic data marked by the red dashed line. Comparing these synthetic data with the observed data and the ground-truth control data, respectively, we observe a perfect match during the first 12 weeks and a close alignment in the second half. The difference between the observed 12-week 1mg data and the red dashed synthetic data after the 12th week can be interpreted as the patient's expected individual treatment effect (eITE) estimated by Equation (18). The critical point is that the synthetic control arm created through the proposed algorithm corresponds individually to the active or crossover arm, forming a synthetic individually controlled arm. Generating and utilizing the synthetic individually controlled arm makes it possible to estimate ITEs or eITEs.

This simple example clearly illustrates a problem that does not have a readily available logical solution despite its apparent simplicity in the reference. For example, look at the blue-colored data. Seemingly, synthetic control methods could create synthetic control data for the data. However, traditional synthetic control methods rely on the availability of control units, which are absent here, to approximate it by taking the weighted sum of them. Thus, this algorithm provides an innovative way to infer these control data by circumventing the need for explicit control units, highlighting its capability to solve problems that traditional algorithms cannot address.

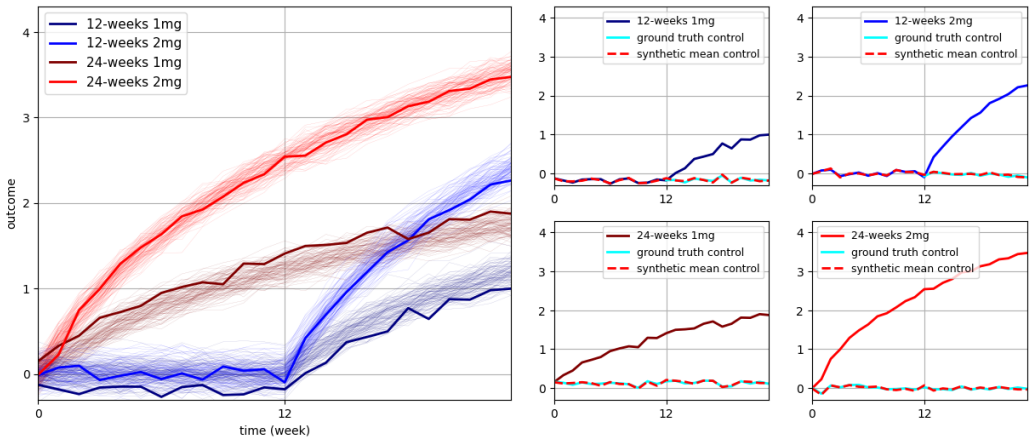

Figure 1: Illustrative crossover trial dataset (linear efficacy) and synthetic control data generation

5.2 SIMULATED DATA

5.2.1 LDL CHOLESTEROL DATA

For quantitative analysis, LDL cholesterol simulated data was utilized. Data for the treatment and control groups were created using the PK/PD model (Faltaos et al., 2006; Yokote et al., 2008; Kim et al., 2011) of statin drugs prescribed to hypercholesterolaemic patients. The experiment was conducted using data with the same characteristics as those generated by SyncTwin (Qian et al., 2021) and SSA-GMM (Ahn & Vashist, 2024).

As explained in Section 2 on the motivation of this paper, the proposed algorithm was developed by converting SSA-GMM into the S-learner framework. While this greatly enhances its versatility by allowing application to various types of treatments, its performance should remain equivalent to the original GMM-based counterfactual generator. However, we achieved better results by finding optimized parameters in the GMCG algorithm. As shown in Table 1 below, the MAE of SSA-GMM reached a level of 0.07, but GMCG was able to reduce the error to a much lower level. When compared to the SyncTwin, SCM (Synthetic Control Method), and CRN (Bica et al., 2020b) models, the error gap becomes even larger.

Table 1: LDL Cholesterol Data Experiment Results

Method	$N = N_0 + N_1 = 200 + 200 = 400$			$N = N_0 + N_1 = 1000 + 200 = 1200$		
	$p_0 = 0.1$	$p_0 = 0.25$	$p_0 = 0.5$	$p_0 = 0.1$	$p_0 = 0.25$	$p_0 = 0.5$
GMCG	**0.042**	**0.038**	**0.043**	**0.040**	**0.039**	**0.035**
SSA-GMM	0.072	0.072	0.076	0.070	0.073	0.069
SyncTwin	0.308	0.150	0.116	0.178	0.106	0.094
SCM	0.341	0.151	0.150	0.231	0.172	0.158
CRN	0.530	0.631	0.343	0.456	0.404	0.371

5.2.2 LDL CHOLESTEROL DATA WITH NO CONTROLS

The reason for using LDL cholesterol data is not just to show superiority over SSA-GMM and the other algorithms. As sketched in the introduction and illustratively shown in Section 5.1, it is intended to be used for more quantitative analysis of crossover trials. However, the composition of the data differs from that in Table 1. In Table 1, $N_0 = 200$ control data and $N_1 = 200$ treatment data were used. In addition, $N_0 = 1000$ control data and $N_1 = 200$ treatment data were also used as training data. But now, we aim to evaluate a much more challenging task: using no control data at all. This is logically unsound because comparing the results of the treatment data with the control data is the golden standard for determining the treatment effect. Without comparison, the effect of the treatment cannot be discussed. Instead, we assumed that we would test the experimental data at various doses. The total duration of the experiment is $T = 30$. Until time $t = 25$, no patient receives the drug. After time $t = 25$, each patient takes a random dose of statin between 0mg and 10mg. This marks a crossover point at time $t = 25$.

After training the model using random doses data, we tested it on the 10mg test data used in Table 1 and obtained the results shown in Table 2. When comparing the results for $N = 1200$, the performance was similar to SCM. Increasing N slightly improved the results. However, the data for $N = 400$ did not train properly. The results in Table 1 were obtained using control data, while the results in Table 2 were obtained by training the model without control data, highlighting the notable performance of GMCG. As explained in the Related Work section, models like SCM and SyncTwin do not function without control data because they create synthetic control by finding linear or nonlinear combinations of control units. Although recurrent network models like CRN can be tested, considering the poor results in Table 1, it is not possible to obtain proper results without training with control data. The results in Tables 1 and 2 are numerical results in terms of MAE, and examples for visual reference are provided in the Appendix D.

GMM-type algorithms use the EM algorithm to find the local maximum of the given likelihood function. The EM algorithm for GMCG training also finds the local maximum of the likelihood function (see Appendix B for the EM algorithm for GMCG). This means that results are influenced

Table 2: LDL Cholesterol Data Experiment Results (No Control Group)

Method	$N = 1200$			$N = 2000$		
	$p_0 = 0.1$	$p_0 = 0.25$	$p_0 = 0.5$	$p_0 = 0.1$	$p_0 = 0.25$	$p_0 = 0.5$
GMCG	0.251	0.199	0.168	0.184	0.150	0.152

by the initial values. In the GMCG algorithm for crossover trials, the choice of initial values affected performance. Applying the EM algorithm from random values may make it difficult to achieve results like those in Table 2. The data structure of crossover trials is fundamentally curved and requires multiple mixture components. The synthetic control data by zero dose is particularly influenced by mixture components derived from low-dose data. Therefore, the GMCG-EM algorithm should be executed using values obtained by first applying a GMM to data created by low doses. The results in Table 2 used two mixture components for dose data between 0mg and 3mg, one for 3mg to 7mg, and one for 7mg to 10mg. Thus, a total of $K = 4$ mixture components were used.

Although the MAE is larger compared to the GMCG results in Table 1, this method has potentially powerful implications. First, to the best of the authors' knowledge, this is the first attempt to solve a causal inference problem without using control data. In the field of clinical trials, there are single-arm trials (Wang et al., 2024), but they make control groups using historical or external controls. However, the GMCG method structurally learns the dose response from the experimental data alone and extrapolates the prediction for zero dose. Second, clinical trials without any control data present a new form of clinical trials that are more economical, easier to recruit patients, and more ethically suitable for patients in many cases where historical or external controls cannot be sufficiently obtained, thus opening up broader possibilities.

6 Conclusion

This paper proposes a counterfactual generator algorithm based on the Gaussian mixture model to create synthetic control arms. It can generate counterfactual data under typical experimental conditions where drug dosages vary continuously over time. We demonstrated this algorithm as a method to generate synthetic placebo arms for clinical trials, such as crossover trials, where a placebo control arm is absent. The methodology of this algorithm has potential applications in various causal inference problems, not only in the healthcare field but also in econometrics and other domains.

REFERENCES

Alberto Abadie. Semiparametric difference-in-differences estimators. *The review of economic studies*, 72(1):1–19, 2005.

Alberto Abadie and Javier Gardeazabal. The economic costs of conflict: A case study of the basque country. *American economic review*, 93(1):113–132, 2003.

Alberto Abadie, Alexis Diamond, and Jens 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.

Jong-Hoon Ahn and Akshay Vashist. A linear algebraic framework for counterfactual generation. In *The Twelfth International Conference on Learning Representations*, 2024.

Scott M Berry, Bradley P Carlin, J Jack Lee, and Peter Muller. *Bayesian adaptive methods for clinical trials*. CRC press, 2010.

Ioana Bica, Ahmed Alaa, and Mihaela Van Der Schaar. Time series deconfounder: Estimating treatment effects over time in the presence of hidden confounders. In *International conference on machine learning*, pp. 884–895. PMLR, 2020a.

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*, 2020b.

Nick Doudchenko, Khashayar Khosravi, Jean Pouget-Abadie, Sebastien Lahaie, Miles Lubin, Vahab Mirrokni, Jann Spiess, et al. Synthetic design: An optimization approach to experimental design with synthetic controls. *Advances in Neural Information Processing Systems*, 34:8691–8701, 2021.

Kerry Dwan, Tianjing Li, Douglas G Altman, and Diana Elbourne. Consort 2010 statement: extension to randomised crossover trials. *bmj*, 366, 2019.

Diana R Elbourne, Douglas G Altman, Julian PT Higgins, Francois Curtin, Helen V Worthington, and Andy Vail. Meta-analyses involving cross-over trials: methodological issues. *International journal of epidemiology*, 31(1):140–149, 2002.

Demiana William Faltaos, Saïk Urien, Valérie Carreau, Marina Chauvenet, Jean Sebastian Hulot, Philippe Giral, Eric Bruckert, and Philippe Lechat. Use of an indirect effect model to describe the ldl cholesterol-lowering effect by statins in hypercholesterolaemic patients. *Fundamental & clinical pharmacology*, 20(3):321–330, 2006.

Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*. Chapman and Hall/CRC, 1995.

Changran Geng, Harald Paganetti, and Clemens Grassberger. Prediction of treatment response for combined chemo-and radiation therapy for non-small cell lung cancer patients using a biomathematical model. *Scientific reports*, 7(1):13542, 2017.

Jeff Heaton. Ian goodfellow, yoshua bengio, and aaron courville: Deep learning: The mit press, 2016, 800 pp, isbn: 0262035618. *Genetic programming and evolvable machines*, 19(1):305–307, 2018.

P Heeson. Crossover trials: what are they and what are their advantages and limitations. *Students 4 Best Evidence*, 2020.

Jimyon Kim, Byung-Jin Ahn, Hong-Seok Chae, Seunghoon Han, Kichan Doh, Jeongeun Choi, Yong K Jun, Yong W Lee, and Dong-Seok Yim. A population pharmacokinetic–pharmacodynamic model for simvastatin that predicts low-density lipoprotein-cholesterol reduction in patients with primary hyperlipidaemia. *Basic & clinical pharmacology & toxicology*, 109 (3):156–163, 2011.

Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences*, 116(10):4156–4165, 2019.

Valentin Letailleur, Isabelle Chaillol, Fanny Cherblanc, Herve Ghesquieres, Frederic Peyrade, Fabrice Jardin, Stephanie Guidez, Fontanet Delices Bijou, Pierre Sesques, Cédric Rossi, et al. Synthetic control arm from clinical trials and real-world data from lysa group for untreated diffuse large b cell lymphoma patients aged over 80 years: A bona fide strategy for innovative clinical trials. *Blood*, 142:72, 2023.

Bryan Lim. Forecasting treatment responses over time using recurrent marginal structural networks. *Advances in neural information processing systems*, 31, 2018.

Geoffrey J McLachlan and Thriyambakam Krishnan. *The EM algorithm and extensions*. John Wiley & Sons, 2007.

Edward J Mills, An-Wen Chan, Ping Wu, Andy Vail, Gordon H Guyatt, and Douglas G Altman. Design, analysis, and presentation of crossover trials. *Trials*, 10:1–6, 2009.

Sarah Jane Nolan, Ian Hambleton, and Kerry Dwan. The use and reporting of the cross-over study design in clinical trials and systematic reviews: a systematic assessment. *PLoS One*, 11(7): e0159014, 2016.

Gabriel Okasa. Meta-learners for estimation of causal effects: Finite sample cross-fit performance. *arXiv preprint arXiv:2201.12692*, 2022.

Philip Pallmann, Alun W Bedding, Babak Choodari-Oskooei, Munyaradzi Dimairo, Laura Flight, Lisa V Hampson, Jane Holmes, Adrian P Mander, Lang'o Odondi, Matthew R Sydes, et al. Adaptive designs in clinical trials: why use them, and how to run and report them. *BMC medicine*, 16:1–15, 2018.

Zhaozhi Qian, Yao Zhang, Ioana Bica, Angela Wood, and Mihaela van der Schaar. Synctwin: Treatment effect estimation with longitudinal outcomes. *Advances in Neural Information Processing Systems*, 34:3178–3190, 2021.

Pedro HC Sant'Anna and Jun Zhao. Doubly robust difference-in-differences estimators. *Journal of econometrics*, 219(1):101–122, 2020.

César Serrano, Sara Rothschild, Guillermo Villacampa, Michael C Heinrich, Suzanne George, Jean-Yves Blay, Jason K Sicklick, Gary K Schwartz, Sameer Rastogi, Robin L Jones, et al. Rethinking placebos: embracing synthetic control arms in clinical trials for rare tumors. *Nature medicine*, 29 (11):2689–2692, 2023.

Uri Shalit, Fredrik D Johansson, and David Sontag. Estimating individual treatment effect: generalization bounds and algorithms. In *International conference on machine learning*, pp. 3076–3085. PMLR, 2017.

Minyan Wang, Huan Ma, Yun Shi, Haojie Ni, Chu Qin, and Conghua Ji. Single-arm clinical trials: design, ethics, principles. *BMJ Supportive & Palliative Care*, 2024.

Yixin Wang and David M Blei. The blessings of multiple causes. *Journal of the American Statistical Association*, 114(528):1574–1596, 2019.

Koutaro Yokote, Hideaki Bujo, Hideki Hanaoka, Masaki Shinomiya, Keiji Mikami, Yoh Miyashita, Tetsuo Nishikawa, Tatsuhiko Kodama, Norio Tada, and Yasushi Saito. Multicenter collaborative randomized parallel group comparative study of pitavastatin and atorvastatin in japanese hypercholesterolemic patients: collaborative study on hypercholesterolemia drug intervention and their benefits for atherosclerosis prevention (chiba study). *Atherosclerosis*, 201(2):345–352, 2008.

Xiner Zhou, Herbert Pang, Christiana Drake, Hans Ulrich Burger, and Jiawen Zhu. Estimating treatment effect in randomized trial after control to treatment crossover using external controls. *Journal of Biopharmaceutical Statistics*, pp. 1–29, 2024.

A NOTATIONS

Numbers and Arrays

a	A scalar (integer or real)
$\boldsymbol{a}$	A vector
$\boldsymbol{A}$	A matrix
$\boldsymbol{I}_n$	Identity matrix with n rows and n columns
$\boldsymbol{I}$	Identity matrix with dimensionality implied by context
$\mathrm{diag}(\boldsymbol{a})$	A square, diagonal matrix with diagonal entries given by $\boldsymbol{a}$
a	A scalar random variable
$\mathbf{a}$	A vector-valued random variable
$\mathbf{A}$	A matrix-valued random variable
$[\boldsymbol{a};\ \boldsymbol{a}] \equiv \begin{bmatrix} \boldsymbol{a} \\ \boldsymbol{a} \end{bmatrix}$	Vectors concatenation
$[\boldsymbol{A};\ \boldsymbol{A}] \equiv \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{A} \end{bmatrix}$	Matrices concatenation

Sets and Graphs

$\mathbb{A}$	A set
$\mathbb{R}$	The set of real numbers
$\{0, 1\}$	The set containing 0 and 1
$\{0, 1, \ldots, n\}$	The set of all integers between 0 and n

Probability and Information Theory

$P(\mathrm{a})$	A probability distribution over a discrete variable
$p(\mathrm{a})$	A probability distribution over a continuous variable, or over a variable whose type has not been specified
$\mathrm{a} \sim P$	Random variable a has distribution P
$\mathbb{E}_{\mathrm{x}\sim P}[f(x)]$ or $\mathbb{E}f(x)$	Expectation of $f(x)$ with respect to $P(\mathrm{x})$
$\mathrm{Var}(f(x))$	Variance of $f(x)$ under $P(\mathrm{x})$
$\mathrm{Cov}(f(x), g(x))$	Covariance of $f(x)$ and $g(x)$ under $P(\mathrm{x})$
$\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	Gaussian distribution over $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$

B EXPECTATION-MAXIMIZATION ALGORITHMS FOR GMCG

For the case of having noise $\boldsymbol{\eta}^{(t)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}^{(t)})$ with a diagonal matrix $\boldsymbol{\Psi}^{(t)}$, Eq. (1) implies a probability distribution over $\boldsymbol{x}^{(t)}$-space for a given $\mathbf{s} \in \mathbb{R}^M$ of the form

$$p(\boldsymbol{x}^{(t)}|\boldsymbol{s}) = |2\pi\boldsymbol{\Psi}^{(t)}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}^{(t)} - \boldsymbol{W}^{(t)}\boldsymbol{s})^T(\boldsymbol{\Psi}^{(t)})^{-1}(\boldsymbol{x}^{(t)} - \boldsymbol{W}^{(t)}\boldsymbol{s})\right\}. \tag{22}$$

With a Gaussian mixture prior over $\boldsymbol{s}$ defined by Eq. (3), we obtain the marginal distribution of $\boldsymbol{x}^{(t)}$ in the form

$$p(\boldsymbol{x}^{(t)}) = \sum_{k=1}^{K} \pi_k \cdot |2\pi\boldsymbol{V}_k^{(t)}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}^{(t)} - \boldsymbol{W}^{(t)}\boldsymbol{\mu}_k)^T\boldsymbol{V}_k^{(t)^{-1}}(\boldsymbol{x}^{(t)} - \boldsymbol{W}^{(t)}\boldsymbol{\mu}_k)\right\}, \tag{23}$$

where the model covariance is $\boldsymbol{V}_k^{(t)} = \boldsymbol{\Psi}^{(t)} + \boldsymbol{W}^{(t)}\boldsymbol{\Sigma}_k\boldsymbol{W}^{(t)T}$.

We can also use the colon notation $\boldsymbol{x} \equiv \boldsymbol{x}^{(:)}$ and $\boldsymbol{W} \equiv \boldsymbol{W}^{(:)}$ defined by Eq. (5). Then, we have the conditional distribution and the marginal distribution in the form of

$$p(\boldsymbol{x}|\mathbf{s}) = |2\pi\boldsymbol{\Psi}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{W}\boldsymbol{s})^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x} - \boldsymbol{W}\boldsymbol{s})\right\} \tag{24}$$

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k \cdot |2\pi\boldsymbol{V}_k|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{W}\boldsymbol{\mu}_k)^T\boldsymbol{V}_k^{-1}(\boldsymbol{x} - \boldsymbol{W}\boldsymbol{\mu}_k)\right\}. \tag{25}$$

where $\boldsymbol{V}_k = \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_k\boldsymbol{W}^T$. By Bayes' rule, this leads to the posterior distribution of the form

$$\begin{aligned} p(\boldsymbol{s}|\boldsymbol{x}) &= \sum_k p(\boldsymbol{s}|\boldsymbol{x}, k)p(k|\boldsymbol{x}) = \sum_k p(k|\boldsymbol{x}) \cdot p(\boldsymbol{x}|\boldsymbol{s})p(\boldsymbol{s}|k)/p(\boldsymbol{x}|k) \\ &= \sum_k p(k|\boldsymbol{x}) \cdot |2\pi\boldsymbol{M}_k|^{-\frac{1}{2}} \exp\{ -\frac{1}{2}\left((\boldsymbol{s} - \boldsymbol{\mu}_k) - \boldsymbol{M}_k\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x} - \boldsymbol{W}\boldsymbol{\mu}_k)\right)^T \\ &\qquad \times \boldsymbol{M}_k^{-1}\left((\boldsymbol{s} - \boldsymbol{\mu}_k) - \boldsymbol{M}_k\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x} - \boldsymbol{W}\boldsymbol{\mu}_k)\right) \} \end{aligned} \tag{26}$$

where $\boldsymbol{M}_k = (\boldsymbol{\Sigma}_k^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W})^{-1}$.

Now we have the joint distribution of observational data $\boldsymbol{x}_n$ and the latent variables $\mathbf{s}_n$ in the form of

$$\begin{aligned} p(\boldsymbol{x}_n, \boldsymbol{s}_n) &= |2\pi\boldsymbol{\Psi}|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}_n - \boldsymbol{W}\boldsymbol{s}_n)^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_n - \boldsymbol{W}\boldsymbol{s}_n)\right\} \times \\ &\quad \sum_{k=1}^{K} \pi_k |2\pi\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{s}_n - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{s}_n - \boldsymbol{\mu}_k)\right\}. \end{aligned} \tag{27}$$

By calculating the expectation values

$$\gamma_{nk} \equiv p(k|\boldsymbol{x}_n) \tag{28}$$

$$\langle \boldsymbol{s}_n \rangle_k = \int p(\boldsymbol{s}|\boldsymbol{x}_n, k)\ \boldsymbol{s}\ |d\mathbf{s}| \tag{29}$$

$$\langle \boldsymbol{s}_n \rangle = \int p(\boldsymbol{s}|\boldsymbol{x}_n)\ \mathbf{s}\ |d\boldsymbol{s}| \tag{30}$$

$$\langle (\boldsymbol{s}_n - \langle \boldsymbol{s}_n \rangle)(\boldsymbol{s}_n - \langle \boldsymbol{s}_n \rangle)^T \rangle = \sum_k p(k|\boldsymbol{x}_n)\ \mathbf{M}_k, \tag{31}$$

we can obtain the expectation of the complete data log-likelihood in the form of

$$\begin{aligned}\langle \mathcal{L}_C \rangle &= \sum_{n=1}^{N}\sum_{k=1}^{K}\gamma_{nk}\{\ln \pi_k - \frac{1}{2}\ln|\boldsymbol{\Sigma}_k| - \frac{1}{2}\mathrm{tr}(\boldsymbol{\Sigma}_k^{-1}\langle \boldsymbol{s}_n\boldsymbol{s}_n^T\rangle) + \boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\langle \boldsymbol{s}_n\rangle - \frac{1}{2}\boldsymbol{\mu}_k^T\boldsymbol{\Sigma}_k^{-1}\boldsymbol{\mu}_k \\ &\quad -\frac{1}{2}\ln|\boldsymbol{\Psi}| - \frac{1}{2}\boldsymbol{x}_n^T\boldsymbol{\Psi}^{-1}\boldsymbol{x}_n + \boldsymbol{x}_n^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\langle \boldsymbol{s}_n\rangle - \frac{1}{2}\mathrm{tr}(\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W}\langle \boldsymbol{s}_n\boldsymbol{s}_n^T\rangle)\} \\ &\quad -\lambda(\sum_{k=1}^{K}\pi_k - 1) \end{aligned} \tag{32}$$

where the expectation values are given by

E-steps:

$$\gamma_{nk} = \frac{\pi_k\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{W}\boldsymbol{\mu}_k, \boldsymbol{V}_k)}{\sum_{k'}\pi_{k'}\mathcal{N}(\boldsymbol{x}_n|\boldsymbol{W}\boldsymbol{\mu}_{k'}, \boldsymbol{V}_{k'})} \tag{33}$$

$$\langle \boldsymbol{s}_n\rangle_k = \boldsymbol{\mu}_k + \boldsymbol{M}_k\boldsymbol{W}^T\boldsymbol{\Psi}^{-1}(\boldsymbol{x}_n - \boldsymbol{W}\boldsymbol{\mu}_k) \tag{34}$$

$$\langle \boldsymbol{s}_n\rangle = \sum_k \gamma_{nk}\langle \boldsymbol{s}_n\rangle_k \tag{35}$$

$$\langle \boldsymbol{s}_n\boldsymbol{s}_n^T\rangle = \sum_k \gamma_{nk}\left(\boldsymbol{M}_k + (\langle \boldsymbol{s}_n\rangle_k - \langle \boldsymbol{s}_n\rangle)(\langle \boldsymbol{s}_n\rangle_k - \langle \boldsymbol{s}_n\rangle)^T\right) + \langle \boldsymbol{s}_n\rangle\langle \boldsymbol{s}_n\rangle^T \tag{36}$$

with $\boldsymbol{V}_k = \boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{\Sigma}_k\boldsymbol{W}^T$ and $\boldsymbol{M}_k = (\boldsymbol{\Sigma}_k^{-1} + \boldsymbol{W}^T\boldsymbol{\Psi}^{-1}\boldsymbol{W})^{-1}$.

Equation (32) is maximized by the following M-step formulas

M-steps:

$$\pi_k = \frac{\sum_n \gamma_{nk}}{N} \tag{37}$$

$$\boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}}\sum_n \gamma_{nk}\langle \boldsymbol{s}_n\rangle \tag{38}$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}}\sum_n \gamma_{nk}(\langle \boldsymbol{s}_n\boldsymbol{s}_n^T\rangle - \boldsymbol{\mu}_k\langle \boldsymbol{s}_n\rangle^T - \langle \boldsymbol{s}_n\rangle\boldsymbol{\mu}_k^T + \boldsymbol{\mu}_k\boldsymbol{\mu}_k^T) \tag{39}$$

$$\boldsymbol{W} = \left[\sum_n \boldsymbol{x}_n\langle \boldsymbol{s}_n\rangle^T\right]\left[\sum_n \langle \boldsymbol{s}_n\boldsymbol{s}_n^T\rangle\right]^{-1} \tag{40}$$

$$\boldsymbol{\Psi} = \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{x}_n\boldsymbol{x}_n^T - 2\boldsymbol{W}\langle \boldsymbol{s}_n\rangle\boldsymbol{x}_n^T + \boldsymbol{W}\langle \boldsymbol{s}_n\boldsymbol{s}_n^T\rangle\boldsymbol{W}^T. \tag{41}$$

C ADDITIONAL MATERIALS FOR CROSSOVER TRIAL EXAMPLE

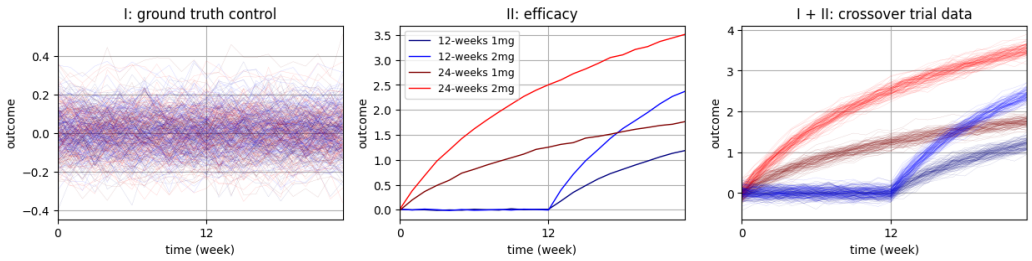

Figure 2: A simple illustrative example of creating crossover trials data.

Data Generation. The linear efficacy data presented in Figure 1 were created using the method shown in Figure 2. First, as illustrated in the left panel, outcome data for 400 patients were created by adding noise mainly within the range of -0.2 to 0.2. Figure 1 used these data as the ground-truth control. The middle panel shows the assumed efficacy for four groups based on the duration and dosage of the drug used. It was assumed that all patients in each group exhibited the same efficacy. The assumed curve shape was generated as a function of time, specifically using $\log(t+1)$ for $0 \leq t < 24$. The two blue-shaded curves are horizontal translations of the two red-shaded curves. The final synthetic crossover trial data, shown in the right panel, were created by combining the data from panels I and II for each patient. This way represents the simplest form of data and does not accurately reflect accurate clinical trial data. Clinical data include discontinuous values, missing values, and various complex covariate structures. While covariate data can provide additional helpful information for experimental results, they were excluded from this example.

Nonlinear Efficacy. In Figure 1, the synthetic control data arm could be created with only two different dose groups per arm due to the assumption that the efficacy is proportional to the dose administered. However, in a more general case, where the relationship between dose and efficacy is nonlinear, the generation of synthetic control data requires more dose groups. Figure 3 presents an example that addresses this question, showing four dose groups for each arm: shades of red representing four dose groups with 0.5, 1.0, 1.5, and 2.0 mg over the full 24-week period, and shades of blue representing four other groups with the same doses during only the last 12 weeks.

The nonlinear relationship assumed here is modeled by $(\text{treatment effect by a dose}) = (\text{dose} + 0.2 \times \text{dose}^2) \times (\text{treatment effect by 1mg dose})$. Each group consists of 100 individuals, totaling 800 training data. The four small plots on the right show synthetic control data for four hypothetical patients taking 1.0 mg and 2.0 mg in each arm, with the generated synthetic mean control data closely approximating the ground truth control.

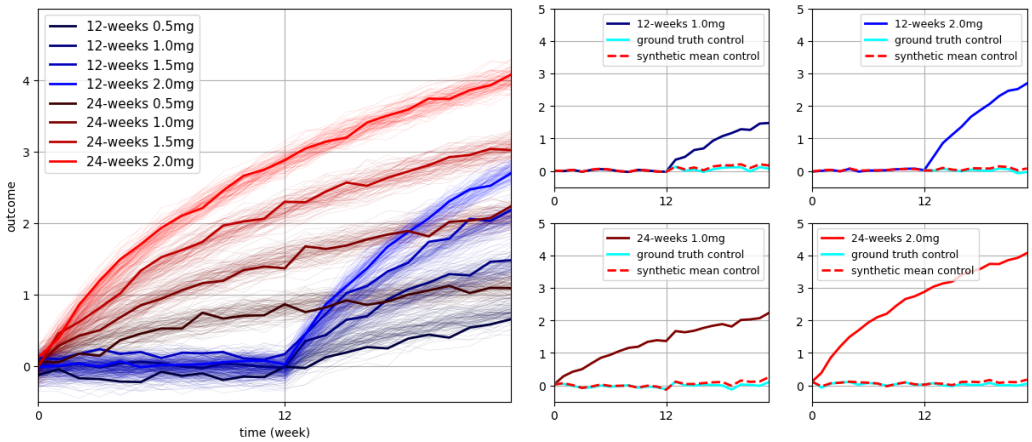

Figure 3: Simulated crossover trial dataset (nonlinear efficacy) and synthetic control data generation.

This problem's nonlinear effect cannot be adequately captured with only two dose groups per arm. The Appendix provides additional results related to this issue. This situation is similar to needing at least three points, rather than just two, to determine a quadratic function's position, direction, and curvature. Therefore, more diverse dose levels are necessary in the experimental data to uncover complex or hidden effects in counterfactual predictions. Therefore, more diverse dose levels are likely necessary in the experimental data to uncover more complex or hidden effects in counterfactual predictions.

This section demonstrates how the proposed GMCG algorithm can be applied to crossover trial data and the types of results it can produce. However, it is important to note that this paper does not claim guaranteed success for all types of crossover trials. As an algorithm within the GMM family, the GMCG algorithm's clear and concise mathematical model highlights its potential to tackle new types of synthetic data generation problems. Discussing this potential alone represents a meaningful achievement.

Non-trivial Control The ground-truth control data in Figure 1 or 2 follows a simple linear trend, which may seem trivial. However, the proposed method does not fit the shape or distribution of the ground-truth control. Figure 3 demonstrates that the ground-truth control data is well predicted even when following a sine wave pattern.

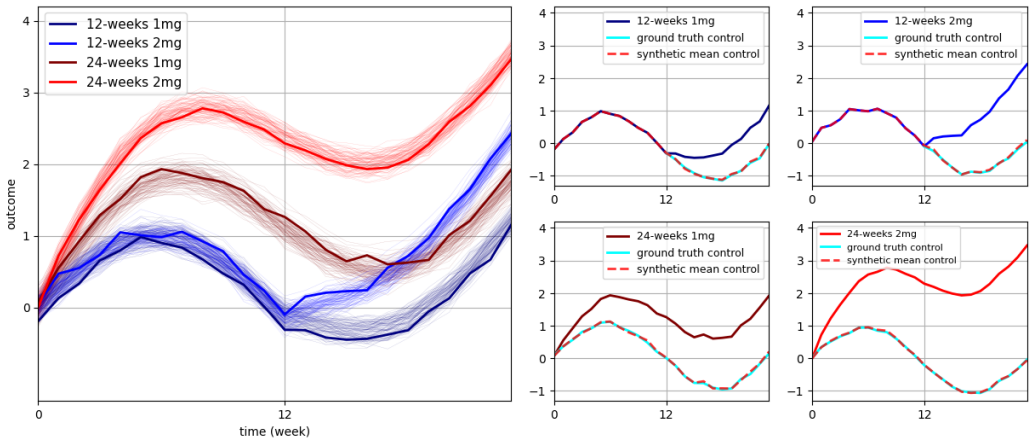

Figure 4: Simulated crossover trial dataset (linear efficacy and sinusodial ground truth control) and synthetic control data generation.

The problem of estimating the nonlinear effect of dosage shown in Figure 3 cannot be adequately addressed with only two dose groups per arm. Therefore, Figure 3 uses four dose groups per arm. Figure 5 shows the bad result when only two dose groups are used. It demonstrates that the synthetic control for the 12-week treatment group is pulled upward due to the nonlinear effect, and the 24-week treatment group shows unstable predictions with significant fluctuations. Increasing the diversity of dosages in the training data, i.e., using more dose groups per arm, can achieve more stable and accurate counterfactual predictions. For instance, if a hypothetical experiment is conducted where each patient receives slightly different dosages, as shown in Figure 6, the results improve even in more complex nonlinearities in efficacy.

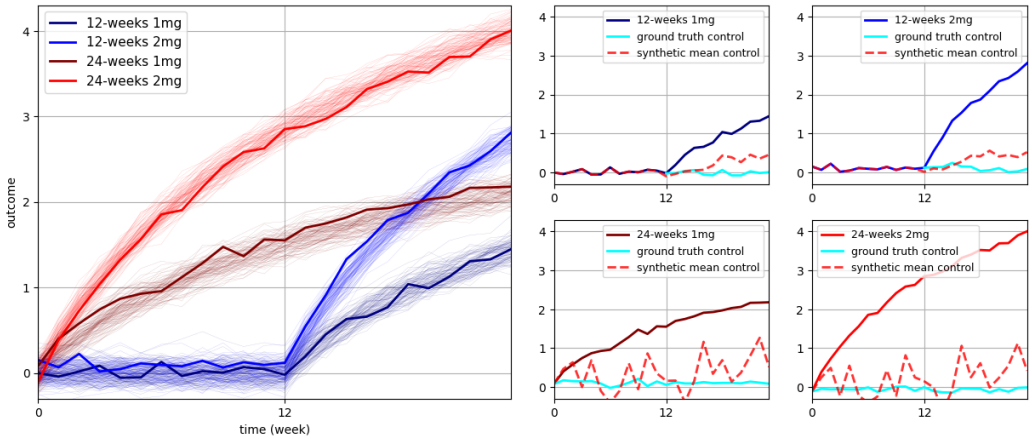

Figure 5: Simulated crossover trial dataset (nonlinear efficacy) and incorrect synthetic control data generation.

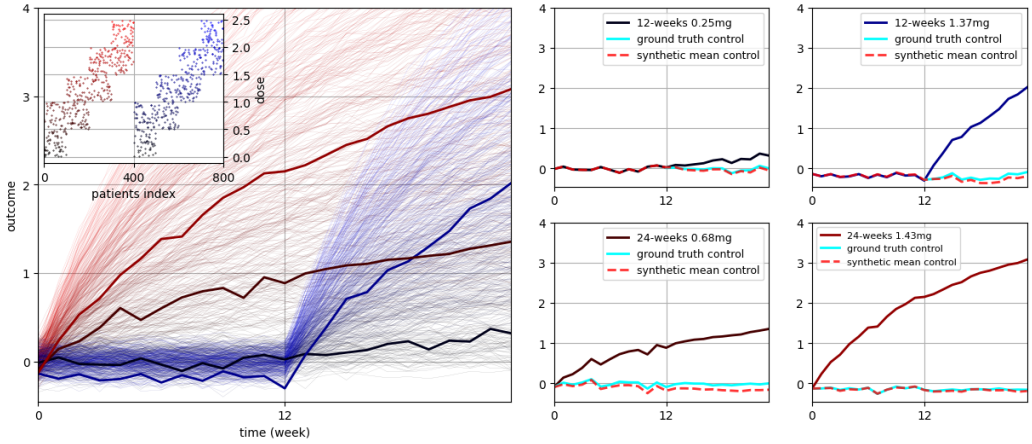

Figure 6: Simulated crossover trial dataset (nonlinear efficacy with diverse doses) and synthetic control data generation.

D ADDITIONAL RESULTS OF LDL DATA EXPERIMENTS

In this experiment, we did not use dimensionality reduction. Thus, $\boldsymbol{W}$ becomes an identity matrix, making the EM algorithm of section B equivalent to the standard GMM algorithm. Specifically, we used one component for each of the treatment and control groups, which trivially means performing single Gaussian density estimation for each group. At this point, there is still an undetermined noise model: $\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I}$ (see Appendix B to check how $\boldsymbol{\Psi}$ affects the parameters of Eq. (16)), and we manually substituted various σ values to obtain the counterfactual predictions given by Eq. (16). The results are shown in Fig. 7. The figure shows that very small or very large sigma values resulted in high error values, while values between 0.1 and 1 yielded the lowest error values. Considering that we added noise with a variance of 0.1 when simulating LDL data, it explains why the lowest error value in terms of MAE was obtained around 0.1.

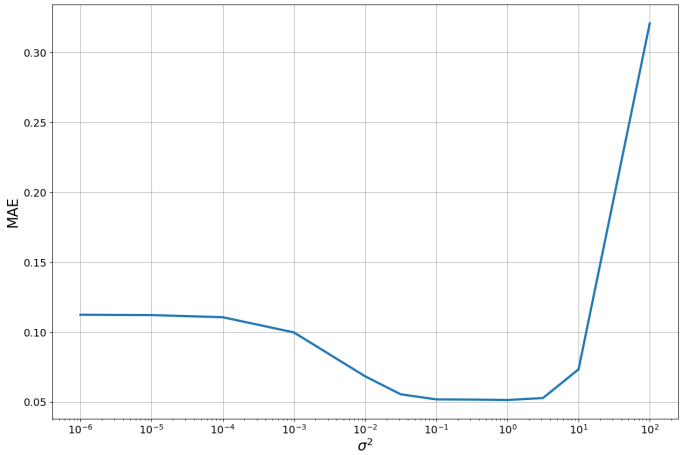

Figure 7: Counterfactual prediction error (MAE) as a function of σ^2 in the noise covariance model $\boldsymbol{\Psi} = \sigma^2 \boldsymbol{I}$.

Figure 8 plots 15 different patients' examples obtained by applying the GMCG algorithm to the LDL cholesterol data. Specifically, for a total time length of $T = 30$, the simulation was performed at $t = 25$ when statin medication was taken, and the factual data, ground truth counterfactual data, and counterfactual prediction obtained through the GMCG algorithm were plotted on each axis. There are two points of observation here: the pretreatment data before $t = 25$ align exactly with the factual data, and the post-treatment data after $t = 25$ closely approximate the ground-truth counterfactual data.

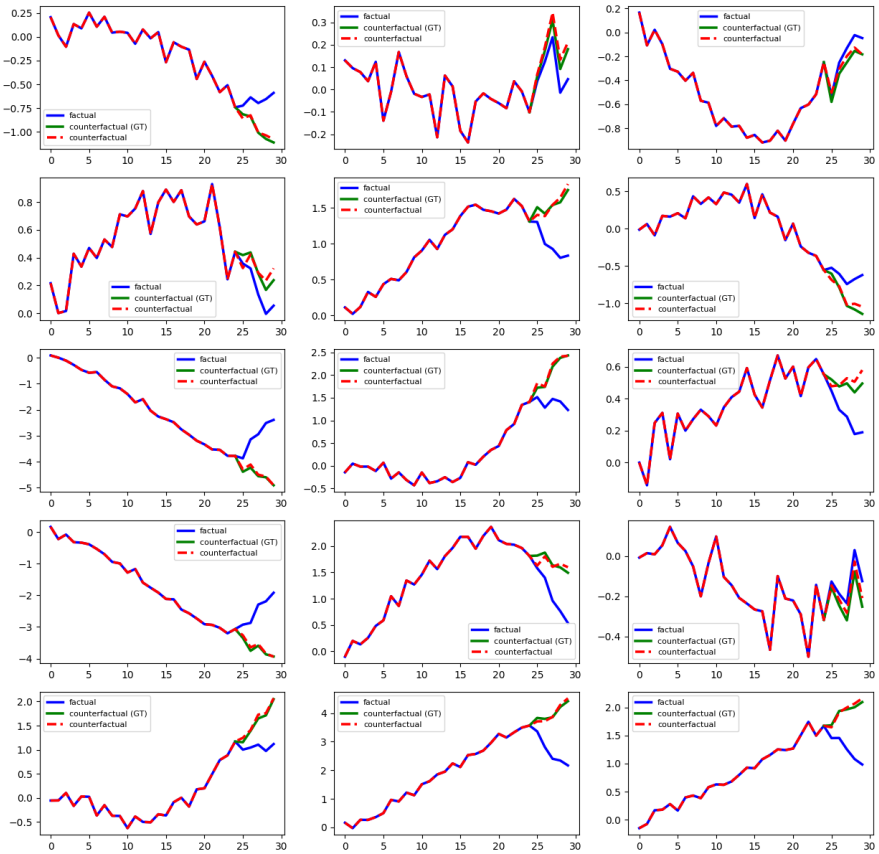

Figure 8: Several plots derived from the results of $p_0 = 0.1$ and $N = 400$ presented in Table 1. Each subplot represents the data of an individual patient. There are two key points to note here: First, the pre-treatment data before $t = 25$ matches exactly between the factual data and counterfactual prediction. Second, the counterfactual prediction after $t = 25$ closely approximates the given ground truth post-treatment data. This serves as further evidence of the excellent results shown in Table 1.

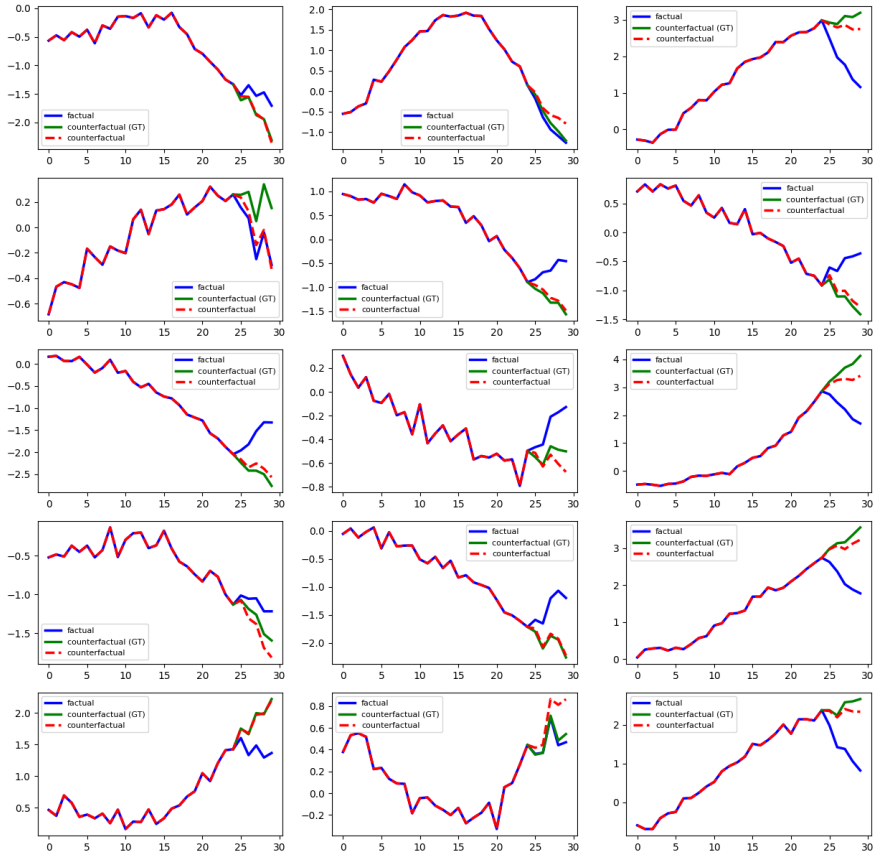

Figure 9: Several plots derived from the results of $p_0 = 0.5$ and $N = 1200$ presented in Table 2. Each subplot represents the data of an individual patient. There are three key points to note here: First, these results were generated by a GMCG model trained without control data, generating control data. Second, the pre-treatment data before $t = 25$ matches exactly between the factual data and counterfactual prediction. Third, the counterfactual prediction after $t = 25$ approximates the given ground truth post-treatment data. This serves as further evidence of the excellent results shown in Table 1.

E SIMULATED TUMOR GROWTH DATA

Although the proposed algorithm is designed to generate synthetic control arms, it can also perform general causal inference, as defined in Section 3.3. This definition aligns with the problem definitions specified in the RMSN and CRN algorithms (Lim, 2018; Bica et al., 2020b). Therefore, in this section, we applied the algorithm to tumor growth data (Geng et al., 2017), which is commonly used for validating general causal inference algorithms. We used data from approximately 10,000 virtual patients and applied it to unbiased and biased data for $T = 10, 20, 30$. The number of mixture components used in the GMCG model is $K = 30$. We compared the state-of-the-art CRN model to the same data and obtained the results. Detailed information is provided in the Appendix.

Table 3 shows the mean absolute error (MAE) obtained by varying δ while keeping τ fixed for data with $T = 10$ and $T = 20$. The error measure was averaged between $t = \tau + 1$ and $t = \tau+\delta$. Surprisingly, the GMM-based GMCG algorithm outperformed the RNN-based CRN regarding MAE. Results for unbiased data are also summarized in the Appendix.

(T, τ)	$(10, 5)$					$(20, 15)$				
δ	1	2	3	4	5	1	2	3	4	5
GMCG	0.0436	0.0484	0.0520	0.0505	0.0497	0.0044	0.0051	0.0054	0.0054	0.0056
CRN	0.0587	0.0596	0.0627	0.0633	0.0644	0.0325	0.0283	0.0283	0.0292	0.0305

Table 3: Mean absolute errors (MAE) comparison for biased tumor growth data.

As the duration T increases, there is a tendency for the error values to decrease. However, this cannot be solely attributed to the more extended data being learned. Fundamentally, with longer durations, the tumor is exposed to more therapy sessions, and the tumor size has already reduced. Consequently, the ground truth counterfactual tumor size for other arbitrary therapy alternatives does not differ significantly from factual tumor size. While the GMCG results closely approximated the ground truth counterfactual data or factual data, the CRN's counterfactual predictions showed some differences from both of them.

How can a GMM-based algorithm yield better results than an RNN-based adversarial network algorithm? Firstly, it is important to recall that GMM can act as an excellent universal approximator as the number of mixture components increases (Heaton, 2018). Additionally, Problem I defined in Section 3.2 is an unsupervised learning problem, and GMM is stably optimized through the EM algorithm (McLachlan & Krishnan, 2007). The most crucial factor lies in the *conditional* probability calculated through Equation (12). Even in regions with sparse training data, the conditional probability calculates a *normalized* probabilistic density distribution conditional on the counterfactual assumption, which creates a robust characteristic against bias. This explains the successful generation of a synthetic control arm in the crossover trials example where training data did not exist.

F ADDITIONAL MATERIALS FOR TUMOR DATA EXAMPLE

F.1 USED HYPERPARAMETERS

To train the CRN for our experiments, we used the default hyperparameters. The exact values used are as follows. These values fall within the optimized hyper-parameters range in Tables 6 and 7 of Appendix J in the CRN paper (Bica et al., 2020b).

The encoder's used hyperparameters:

$$\begin{aligned}&\text{num_epochs} = 100\\&\{\ \text{rnn_hidden_units} : 24,\\&\quad \text{br_size} : 12,\\&\quad \text{fc_hidden_units} : 36,\\&\quad \text{learning_rate} : 0.01,\\&\quad \text{batch_size} : 128,\\&\quad \text{rnn_keep_prob} : 0.9\ \}\end{aligned}$$

The decoder's used hyperparameters:

$$\begin{aligned}&\text{num_epochs} = 100\\&\{\ \text{rnn_hidden_units} : 12,\\&\quad \text{br_size} : 18,\\&\quad \text{fc_hidden_units} : 36,\\&\quad \text{learning_rate} : 0.001,\\&\quad \text{batch_size} : 1024,\\&\quad \text{rnn_keep_prob} : 0.9\ \}\end{aligned}$$

where (br_size) is (balancing representation size) and (rnn_keep_prob) = 1 - (RNN dropout probability).

F.2 ADDITIONAL TABLE

(T, τ)	$(10, 5)$					$(20, 15)$				
δ	1	2	3	4	5	1	2	3	4	5
GMCG	0.0548	0.0636	0.0697	0.0694	0.0682	0.0057	0.0061	0.0071	0.0071	0.0073
CRN	0.0831	0.0795	0.0773	0.0788	0.0776	0.0429	0.0390	0.0348	0.0327	0.0296

Table 4: Mean absolute errors (MAE) comparison for unbiased tumor growth data for $T = 10$ and $T = 20$.

(T, τ)	$(30, 15)$					$(30, 25)$				
δ	1	2	3	4	5	1	2	3	4	5
GMCG	0.0075	0.0072	0.0077	0.0079	0.0081	0.0015	0.0017	0.0019	0.0019	0.0020
CRN	0.0179	0.0226	0.0262	0.0293	0.0321	0.0128	0.0151	0.0167	0.0192	0.0209

Table 5: Mean absolute errors (MAE) comparison for unbiased tumor growth data for $T = 30$.