InterpTwin: An Interpretative Reformulation of SyncTwin

Jingxiao J. Xu
University of Cambridge
jx283@cam.ac.uk

Abstract

The task of treatment effect estimation and inference is a well-known and important problem in the field of medicinal research. Electronic health records (EHR) often provide data from longitudinal settings, with each patient's covariates and outcomes recorded over time before and after treatment interventions. The subject of this paper, InterpTwin, demonstrates a synthetic control method in estimating treatment effects that mimics its predecessor, SyncTwin, while furthermore producing more human-interpretable post-inference results. We identify areas of the SyncTwin model that lack robustness - specifically on the nature of its generated latent vectors - and propose amendments both to its training procedure and model pipeline. This in turn creates a better foundation for which synthetic contributors or 'donors' of certain individuals are found, and their contributing effects better quantified under a more linearly regularised latent space. We reason theoretically why our changes aid the practical use of the model output to domain experts in data-point interpretability, and produce results that show this on simulated data.

1 Introduction

The focus of this paper revolves around *SyncTwin: Treatment Effect Estimation with Longitudinal Outcomes* from Qian et al. [20]. Since we mention their work frequently throughout, from henceforth we shall refer to it as simply 'SyncTwin'.

The fundamental problem of causal inference makes individualised treatment effect (ITE) calculation a non-trivial and well-sought after goal for applied statisticians, particularly in the field of healthcare. Electronic health records (EHR) has given access to covariate data of patients over time, tracking their relevant individual attributes in spans of periods both before and after a treatment intervention. This makes such observational data an appealing alternative data source to randomised control trials (RCTs) for estimating treatment effects [26, 23, 29]. Although various methods in various settings exist in longitudinal treatment effect estimation, we restrict ourselves to the same setting as SyncTwin: Longitudinal and Irregularly sampled data with Point treatment setting, or referred to as *LIP*. In particular the SyncTwin model is able to deal with data that are not sampled at regular intervals, which is the case in many real-life data-collection. For example, heart rate statistics may be recorded more frequently when a patient's condition becomes more critical [19]. In general, there are also more instances outside healthcare that result in sparse and irregularly-sampled data [24, 10].

Characteristically SyncTwin belongs to a family of treatment effect estimation methods which practice 'synthetic control'. These techniques operate on the assumption that counterfactual outcomes lie in the space spanned by factual outcomes of the corresponding treatment; that is, we may look in the pool of factual data corresponding to a certain treatment (known as 'donors') for 'contributors' to linearly generate the counterfactual. As a bonus to treatment effect estimation, we may suppose some

Preprint. Under review.

sort of relationship between an individual and their contributors. As Shi et al. [25] details,

$$\hat{b}_i = \arg\min_b \sum_{t\in\tau}(Y_{it} - \sum_{j\in\mathcal{D}} Y_{jt}b_j)^2 + \Gamma(b) \tag{1}$$

where $\hat{b}_i \in \mathbb{R}^{|\mathcal{D}|}$ are the synthetic weights for individual i, Y_{it} denotes the outcome measurement of individual i at time t, $\mathcal{D}$ is the set of donors, τ is the set of measurement times and Γ represents a restriction function on the values that b may take.

SyncTwin most notably performs this minimisation task in equation (1) within the latent space of the covariates where an established data generating process (DGP) assumes a (linear) latent factor model commonly used in Econometrics [1]. SyncTwin's novelty comes from modelling these latent factors in a non-linear way (in general) by means of auto-encoders. However, our primary focus is to question the robustness of this method, and specifically address the problems it causes down the line with its practical implementation and interpretability.

One of the main aims InterpTwin ('Interpretable-Twin'), the proposal of this paper, is to provide a linearly regularised latent space such that the weighting vectors $\hat{b}_i$ are as linearly expressive as possible and resemble closely to that of regression coefficients. This goal aids linear interpolation and extrapolation of the latent space, moreover the ability to recognise 'anti-similarity' of donors. In particular we permit $\hat{b}_i$ to have negative components, an aspect which SyncTwin does not allow, to deduce from it a better gauge of the common characteristics that shortlisted candidates share. We expect that overall InterpTwin may better bridge the gap between black-box machine learning methods and explainable AI in the field of healthcare [18] and thus act as a more reliable agent for decision making.

Contributions. We introduce concepts that modify SyncTwin in several ways, from changing it's training process to allowing for negative weighting during synthetic matching. We reason why these changes give a more interpretable output of the produced synthetic weights, while at the same time do little to no compromise of SyncTwin's original purpose of treatment effect estimation. We demonstrate such claims on a generated dataset, as well as give measures where better 'interpretability' of InterpTwin is shown.

2 Problem setting

2.1 Notation

The type of data we deal with as already detailed by LIP consists of longitudinal measurements of an individual's attributes (covariates) as well as outcomes (variable of prediction interest) over irregular times both before and after a treatment intervention. We shall follow much of notation convention from SyncTwin.

We consider an observational study with each participant labelled $i \in [N]$. Let the set of control individuals be $\mathcal{I}_0 \subset [N]$, the treated individuals $\mathcal{I}_1 \subset [N]$ and N_0 and N_1 be the sizes of these sets respectively. We denote the times in which pre-treatment observations are made $\mathcal{T}^-$, post-treatment observations $\mathcal{T}^+$ and let the union $\mathcal{T} \equiv \mathcal{T}^- \cup \mathcal{T}^+$. Let the random variable of outcomes at time t for individual i be $y_{it} \in \mathbb{R}$ for $t \in \mathcal{T}$ and similarly the covariates $\mathbf{x}_{it} \in \mathbb{R}^D$. Let $\mathbf{X}_i \in \mathbb{R}^{|\mathcal{T}^-|\times D}$ be the pre-treatment covariate matrix. We shall also adopt Rubin's counterfactual framework [28] the notation $y_{it}(0)$ and $y_{it}(1)$ for the potential outcomes with or without (a binary, indicator) treatment intervention. We refer to vectors $\mathbf{y}_i(a) \equiv (y_{it}(a))_{t\in\mathcal{T}}$, $\mathbf{y}_i^-(a) \equiv (y_{it}(a))_{t\in\mathcal{T}^-}$, $\mathbf{y}_i^+(a) \equiv (y_{it}(a))_{t\in\mathcal{T}^+}$ for some treatment $a \in \{0, 1\}$.

2.2 Assumptions

Similar to SyncTwin we keep the assumptions of (1) Stable Unit Treatment Value Assumption (SUTVA) [28], (2) no anticipation [4] and most importantly (3) the data generating process (DGP) of the outcomes follow the latent factor model:

$$y_{it}(0) = \mathbf{q}_t^T \mathbf{c}_i + \varepsilon_{it} \qquad \forall t \in \mathcal{T} \tag{2}$$

where $\mathbf{c}_i \in \mathbb{R}^K$ is the latent representation of individual i, $\mathbf{q}_t$ is a vector that summarises time depencies of the outcome and $\varepsilon_{it} \in \mathbb{R}$ describes the noise. This data generating process is a strong statement and is the pivotal assumption which motivates SyncTwin's methodology and in turn InterpTwin. In this paper we shall also work under the assumption that data is complete.

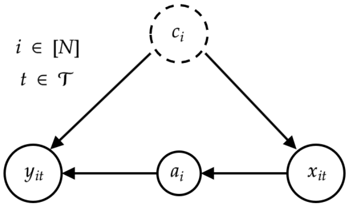

Figure 1: DAG assumed in our model.

SyncTwin presented the $\mathbf{c}_i$ vectors as (hidden) confounding effects of the covariates $\mathbf{X}_i$ and outcomes $\mathbf{y}_i$. During the simulation study conducted in SyncTwin, it adheres to this by including the outcomes $\mathbf{y}_i$ as part of the covariates that estimate $\mathbf{c}_i$ - but this raises some concerns with the DGP assumption in mind (see Appendix A.1). In this paper we will gloss over this and instead view the $\mathbf{c}_i$ vectors as latent representations of our covariate data $\mathbf{X}_i$ only.

Remark. Regardless of whether we treat $\mathbf{c}_i$ as confounding effects or not, we know that $\mathbf{c}_i \perp\!\!\!\perp a_i \mid \mathbf{X}_i$ (by considering the single world intervention graph [21]) and by the no anticipation assumption we must also have the same governing DGP for $y_{it}(1)$ for $t \in \mathcal{T}^-$.

2.3 Motivation: question of interpretability

Aside from the primary goal of a method that accurately estimates ITE (which SyncTwin does well already), a bonus feature that comes along with creating a 'synthetic twin' for individual i is identifying a shortlist of individuals that are 'related'. SyncTwin does this by calculating a set of weights $\mathbf{b}_i$ from a problem modified from equation (1) with estimated latent vectors $\tilde{\mathbf{c}}_i$ instead:

$$\min_{\mathbf{b}} \left\| \tilde{\mathbf{c}}_i - \textstyle\sum_{j \in \mathcal{I}_0} b_j \tilde{\mathbf{c}}_j \right\|^2 \qquad \text{s.t. } b_j \geq 0 \, \forall j \qquad \sum_{j \in \mathcal{I}_0} b_j = 1 \tag{3}$$

It should be noted that on top of DGP, this inherently places a stronger assumption than classically of synthetic control methods (see Appendix A.2). It is also working under the setting where we wish to estimate counterfactual values of the *treated* outcomes. To estimate the counterfactual values of the control outcomes we simply take $\mathbf{c}_j$ from the pool of donors $j \in \mathcal{I}_1$ instead.

The idea is to interpret these b_j weightings as how representative our observations for patient j is of i. After shortlisting 'contributors' with significant b_j, a domain expert may wish to perform further investigation on shortlisted individuals and perhaps find deeper insight to the relationship with outcomes $\mathbf{y}$ within the group. The usefulness of such weights b_j for the domain expert depends on the following three criteria, of which InterpTwin identifies and addresses:

- **Consistency.** Given that in representing i we shortlist individual j, does representing j also shortlist individual i?
- **Validity.** After obtaining a shortlisted sample for i via method 3, do all members of the shortlist *truly* share a common trait or (with what probability) are some members included in by coincidence?
- **Transparency.** What significance are the calculated weights b_j? For instance, how do we interpret $b_j = 0.3$ against $b_j = 0.6$ or anywhere in between? Is the variation human-interpretable?

3 Proposed method: InterpTwin

We first make a statement about InterpTwin's optimality. Due to time, resource and computational constraints many of the hyper-parameters and much of the model's architecture remains un-investigated and not fine-tuned for optimal results. The work we propose serves only as proof-of-concept which is agnostic to varying architectures and set-ups.

3.1 Robustness of latent vector representation

The training loss proposed by SyncTwin in fitting the auto-encoder for estimating counterfactual values of *treated outcomes* (i.e. we are estimating $\mathbf{y}_i^+(0)$ for $i \in \mathcal{I}_1$) consists of

$$\lambda_s \cdot \underbrace{\sum_{i \in \mathcal{I}_0} \left\| \tilde{\mathbf{y}}_i^+(0) - \mathbf{y}_i^+(0) \right\|^2}_{\mathcal{L}_s(\mathcal{I}_0)} + \lambda_r \cdot \underbrace{\sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_1} \left\| \tilde{\mathbf{X}}_i - \mathbf{X}_i \right\|^2}_{\mathcal{L}_r(\mathcal{I}_0 \cup \mathcal{I}_1)} \tag{4}$$

with $\mathcal{L}_s$ corresponding to the supervised loss and $\mathcal{L}_r$ the reconstruction loss of the auto-encoder (Frobenius norm taken). Here $\tilde{\mathbf{y}}_i^+(0) \equiv \tilde{\mathbf{Q}}\tilde{\mathbf{c}}_i$ where $\tilde{\mathbf{Q}}$ has rows of estimates $\mathbf{q}_t$ for $t \in \mathcal{T}^+$, and thus respecting the DGP assumption. However by the remark in section 2.2 the $\mathbf{c}_i$ are independent of treatment intervention, thus it is worrying to train such latent vectors on a loss that is not symmetric in $a_i \in \{0, 1\}$. More explicitly, if we were to further train the same dataset to estimate counterfactual values of *control outcomes* (the $\mathbf{y}_i^+(1)$ for $i \in \mathcal{I}_0$), we cannot hope to obtain the same sets of latent vectors $\{\tilde{\mathbf{c}}_i\}_{i \in [N]}$ for the same individuals. Indeed performing an empirical run-through on simulated data (see section 5) we do observe significant discrepancies (see Appendix A.3).

There is a simple fix to this, which is to make the training loss (4) symmetric. InterpTwin modifies the supervised loss to include treated factuals too:

$$\mathcal{L}_s^* \left(\mathcal{I}_0 \cup \mathcal{I}_1\right) = \sum_{i \in \mathcal{I}_0} \left\| \tilde{\mathbf{y}}_i^+(0) - \mathbf{y}_i^+(0) \right\|^2 + \sum_{i \in \mathcal{I}_1} \left\| \tilde{\mathbf{y}}_i^+(1) - \mathbf{y}_i^+(1) \right\|^2 \tag{5}$$

Remark. The augmented loss (5) makes the further assumption of extending DGP to counterfactuals of the control group $y_{it}(1) = \mathbf{q}_t^T \mathbf{c}_i + \varepsilon_{it}$ for all $t \in \mathcal{T}^+$. This is a very small leap forward from the original DGP assumption so we shall proceed with it.

3.2 Faithfulness to DGP

If indeed our observed data truly follows the latent factor model proposed by the DGP, SyncTwin's training loss (4) doesn't fit the latent vectors to respect this for times $t \in \mathcal{T}^-$. This may induce overfitting to the post-treatment data, discrediting results from ITE estimation. We propose in InterpTwin to add another additional term to the supervised loss which includes pre-treatment outcomes.

$$\mathcal{L}_s^{**} \left(\mathcal{I}_0 \cup \mathcal{I}_1\right) = \mathcal{L}_s^* \left(\mathcal{I}_0 \cup \mathcal{I}_1\right) + \sum_{\mathcal{I}_0 \cup \mathcal{I}_1} \left\| \tilde{\mathbf{y}}_i^- - \mathbf{y}_i^- \right\|^2 \tag{6}$$

By no anticipation, $\mathbf{y}_i^- = \mathbf{y}_i^-(0) = \mathbf{y}_i^-(1)$. In practice, we do not expect changes in (6) or (5) to give better results (both in ITE estimation and interpretation) as we cannot guarantee the truthfulness of our DGP assumption. It does however balance out biases towards fitting to only post-treatment data.

Remark. In using this modified training loss $\mathcal{L}_s^{**}$ we disqualify ourselves from one of the utilities that SyncTwin originally provided - individualised error bounds. By training on the pre-treatment outcome loss explicitly, using it for error bounds would be equivalent to using training loss to predict inference properties.

3.3 Latent space regularisation

One of the distinguishing features of SyncTwin from other synthetic control methods is to assume a non-linear relationship between the covariates $\mathbf{X}_i$ and its latent vector $\mathbf{c}_i$. In this way the $\mathbf{c}_i$ may be modelled as the latent vectors of an auto-encoder structure. This is a weaker assumption than linearity used by Abadie et al. [2] - and indeed would produce a better fitting - but at an interpretability cost. Indeed in general for a non-linear encoding function $\mathbf{c}_i = f_e(\mathbf{X}_i, \mathcal{T}_i^-)$, let us suppose 'similar' individuals demonstrate a linear correlation via $\mathbf{X}_i = \alpha \mathbf{X}_j + \gamma \mathbf{X}_k$ for some constants α and γ (for justification on how this satisfies the criteria in section 2.3, see Appendix A.4). We cannot possibly assert that a similar linear relationship exists for the transformed $\mathbf{c}_i = f_e(\mathbf{X}_i, \mathcal{T}_i^-)$, $\mathbf{c}_j = f_e(\mathbf{X}_j, \mathcal{T}_j^-)$ and $\mathbf{c}_k = f_e(\mathbf{X}_k, \mathcal{T}_k^-)$. Furthermore even if some pseudo-linear relationship exists (from perhaps a loss-based regression method) there are no guarantees that the fitted coefficients are of the same nature as α and γ (e.g. same sign). This violates all three criteria established in section 2.3.

A linear f_e would solve such issue. Indeed taking the same example, preservation of linear transformations means we arrive at exactly the same constants α and γ from regressing $\mathbf{c}_i$ on $\mathbf{c}_j$ and $\mathbf{c}_k$. On the other hand a linear f_e would, on top of the DGP assumption, collapse the problem to an uninteresting linear regression of $\mathbf{y}_i$ on $\mathbf{X}_i$, which eliminates all flexibility that SyncTwin encapsulated in the latent vectors.

InterpTwin bridges the extremes of these two methods by keeping the same non-linear framework SyncTwin provides, but also at the same time *regulating* the latent space - which the $\mathbf{c}_i$ belong - to not 'deviate' greatly from a linear, vector space. In this way we hope to produce more interpretable weights $\mathbf{b}_i$ but also compromise little in ITE estimation accuracy.

We achieve this by introducing a 'helper' term in the training loss which mimics the reconstruction loss had we used a *linear* encoder. A linear encoder exactly achieves the desired regulation, so training with this helper loss encourages InterpTwin's f_e to not deviate far from a linear transformation. Let the linear encoder be denoted as h_e and InterpTwin's decoder be f_d. Our proposed training loss $\mathcal{L}$ is given by

$$\mathcal{L} = \lambda_s \cdot \mathcal{L}_s^{**} + \lambda_r \cdot \mathcal{L}_r + \lambda_h \cdot \underbrace{\sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_1} \left\| f_d(h_e(\mathbf{X}_i, \mathcal{T}_i^-)) - \mathbf{X}_i \right\|^2}_{\mathcal{L}_h(\mathcal{I}_0 \cup \mathcal{I}_1)} \tag{7}$$

with hyper-parameter λ_h representing the importance we give to the helper loss. To aid convergence we alternate the parameters in which we train at a time. A training epoch consists of two stages: 1. training on $\mathcal{L}$ keeping h_e as a fixed function followed by 2. training on $\mathcal{L}$ keeping everything except h_e fixed. The second stage is equivalent (we assume f_d is continuous as it often is in deep neural nets) to training with the loss function

$$\mathcal{L}_{\text{alt}} = \sum_{i \in \mathcal{I}_0 \cup \mathcal{I}_1} \left\| h_e(\mathbf{X}_i, \mathcal{T}_i^-) - f_e(\mathbf{X}_i, \mathcal{T}_i^-) \right\|^2 \tag{8}$$

where f_e is fixed in the above setting. For discussion on how we encode covariates $\mathbf{X}_i$ along with corresponding times $\mathcal{T}_i^-$ in a linear manner via h_e, see Appendix A.5.

3.4 Weighting as linear coefficients

Under regularisation of the latent space from previous section 3.3, we find greater linear interpretability in the weights $\mathbf{b}_i$ found by the matching process (3). Indeed any linear relationship between latent vectors $\mathbf{c}_i$ perhaps approximately translates to linear relationships between the corresponding covariates $\mathbf{X}_i$.

To accommodate for this nicely, InterpTwin gets rid of restrictions $b_j \geq 0$ (non-negativity) and $\sum_j b_j = 1$ (adding up). These constraints are semantic to interpretability and by no means universal as demonstrated in other works [11, 22]. A drawback of SyncTwin is that the weights identified have no way of detecting 'dissimilarity' - individuals who have the opposite trait to that of i. With the new framework, these dissimilar subjects correspond to the weighting vector $b_j < 0$, thus allowing for extrapolation of covariate data.

Furthermore, we have reason to treat b_j as linear coefficients, telling us how $\mathbf{X}_i$ (close to) linearly varies with $\mathbf{X}_j$. This improves on the transparency criterion in section 2.3. With this in mind, InterpTwin replaces SyncTwin's gradient-based approach (for finding weight values) with a much simpler, l_1 norm penalisation via the Lasso [27]. This way we preserve sparse parameter selection properties (encouraging fewer non-zero weights) while relinquishing adding up and non-negativity assumptions.

4 Related Works

Latent space regularisation (LSR) is a well studied topic in machine learning models as it is not only useful in producing meaningful results [17] but also interpreting them [5]. That having said, much of its work relies on knowing a pre-defined set of 'useful' properties of the data, before modifying the training procedure to keep these properties separate and regularised in the latent space [16, 9, 12].

For our case, the identification of groups of individuals that share some common correlations cannot leverage these works, as the nature of common correlations are unknown (and indeed, require further investigation under supervision of domain experts). We insist further that the latent space be *linearly* regularised relative to the encoder, which is rarely desired [15] as often more straightforward techniques such as principal component analysis may be employed. Heinze-Deml et al. [13]'s work most closely matches our regularisation, and indeed served as inspiration for InterpTwin. The Latent Linear Adjustment Auto-encoder [13] auto-encodes the outcome variable (in their work, spatially recorded precipitation data) such that the latent space may be linearly regressed onto by the covariates (circulation data). It does this in a similar training procedure described in section 3.3. However InterpTwin differs by instead focusing on the latent space of the covariates $\mathbf{X}_i$ rather than outcomes, and linking them via DGP assumption.

Other synthetic control methods have also relaxed the additive and non-negative assumptions on the weights $\mathbf{b}_i$. Doudchenko and Imbens [11]'s use of the elastic net [30] enforces restrictions on the magnitude of weight vectors via l_2 penalty. We have no reason to believe in our work that weight magnitudes need to be shrunk towards zero, thus do not inherit the l_2 loss. Abadie et al. [3] allows for negativity of weights, but still requires 'adding-up'. It is discussed by Abadie [1] the risks of allowing for negative weights - that it permits extrapolation of data. Amjad et al. [6] argues that since the objective is to produce accurate predictions, negative weights should be allowed and a linear regression stance should be taken. InterpTwin also adopts this notion, although differs in its work in that we use an auto-encoder instead of singular value thresholding (SVT) methods to denoise the covariates [6].

5 Experiments

We evaluate InterpTwin on the same generated dataset used to test SyncTwin. We first compare as a sanity check the ITE estimation results from InterpTwin and show that the results are more than comparable than that of SyncTwin. We then perform further simulation tests on artificially generated individuals to identify differences of linear latent space regularisation between the two models and how that affects the criteria in section 2.3.

Data Generation. The data which we perform our simulation study surrounds LDL cholesterol levels (our outcome $\mathbf{y}_i$) correlating with covariates (our $\mathbf{X}_i$) which include relevant physiological attributes like serum creatinine, uric acid, serum creatine phosphokinase (CPK), and glycaemia. We used a dataset of $N_0 = N_1 = 200$ individuals with varying levels of (what previous authors defined as) 'confounding bias' probability p. Since the exact data generating setup is used in SyncTwin, to avoid repetition please refer to Qian et al. [20]'s paper for further details.

Benchmarks. Since we borrow off the work of SyncTwin, it is natural to compare with it. As far as we are aware there are no works on the interpretability of sythnetic control weights to the nature we have, so we shall solely compare with SyncTwin. Given our contributions in section 3, we present for testing several ablated versions of InterpTwin which do/do not feature training losses $\mathcal{L}_s^*$, $\mathcal{L}_s^{**}$, $\mathcal{L}_h$ as well as whether the Lasso method was used for the matching process (3) (or whether the original, gradient-based method from SyncTwin was used).

Model hyper-parameters. Notable hyper-parameters worth adjusting within InterpTwin include λ_s, λ_r, λ_h, the dimension number of the $\mathbf{c}_i \in \mathbb{R}^K$ and the 'alpha', α, parameter used for the Lasso objective [27]. If optimisation was priority, these parameters would be subject to further investigation (e.g. cross-validation). However that is not the main purpose here; hyper-parameters in this project were chosen so at first glance they produce sensible results and nothing more. Throughout our experiments, $\alpha = 0.01, \lambda_s = 1, \lambda_r = 50, \lambda_h = 30, K = 40$.

5.1 ITE estimation results

The results in table 1 show comparable results between SyncTwin and all ablated versions of InterpTwin. Indeed, InterpTwin actually slightly outperforms in some setups, perhaps most significantly with SyncTwin + $\mathcal{L}_h$. This phenomenon (if significant) is intriguing, as our theory could no way have predicted this, although may be just a property of this simulated dataset. Nevertheless explaining this

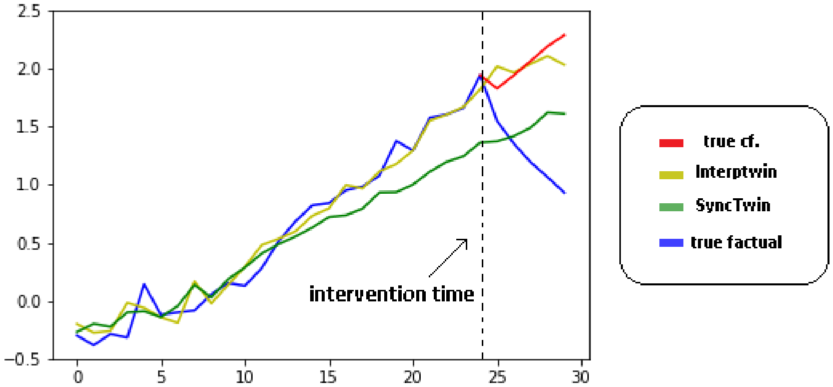

Figure 2: Graph showing synthetically generated 'twins' from both SyncTwin (green) and InterpTwin (yellow) that estimate the true counterfactual (red).

Table 1: Comparison of mean absolute error (MAE) of estimated treatment effect for SyncTwin and all ablated versions of InterpTwin.

Model		MAE		
Components	Lasso?	$p = 0.5$	$p = 0.25$	$p = 0.1$
SyncTwin	No	0.132 (0.010)	0.184 (0.012)	0.194 (0.012)
SyncTwin + $\mathcal{L}_s^*$	No	0.134 (0.010)	0.184 (0.014)	0.192 (0.012)
	Yes	0.115 (0.007)	0.167 (0.011)	0.179 (0.011)
SyncTwin + $\mathcal{L}_h$	No	0.115 (0.008)	0.138 (0.008)	0.173 (0.012)
	Yes	0.122 (0.008)	0.130 (0.007)	0.163 (0.010)
SyncTwin + $\mathcal{L}_s^*$ + $\mathcal{L}_h$	No	0.127 (0.009)	0.158 (0.011)	0.178 (0.012)
	Yes	0.122 (0.008)	0.150 (0.009)	0.173 (0.011)
SyncTwin + $\mathcal{L}_s^{**}$	No	0.135 (0.010)	0.187 (0.013)	0.200 (0.013)
	Yes	0.115 (0.007)	0.181 (0.011)	0.183 (0.011)
SyncTwin + $\mathcal{L}_s^{**}$ + $\mathcal{L}_h$ (InterpTwin)	No	0.128 (0.009)	0.161 (0.011)	0.183 (0.012)
	Yes	0.124 (0.008)	0.151 (0.009)	0.166 (0.010)

is superficial, and a sanity check that InterpTwin does not suffer in its ITE estimation capabilities compared to SyncTwin suffices.

5.2 Validity testing

We focus now on testing the validity criterion proposed in section 2.3. Given the individuals a synthetic control method identifies as 'significiant' in contributing to $i \in [N]$, how many of these relationships are truly existent and not due to spurious correlations? In theory, as the number of individuals grow larger (control and treated), an inevitable consequence is the increase in dimension of the null space spanned by the latent vectors $\mathbf{c}_i$, and thus spurious solutions produced by the matching problem (3) increase in probability. InterpTwin's solution to dealing with this problem is to recognise that with a (close to) linear latent space, any matching identified is useful - by invariance of linear relationships. Take the instance $\mathbf{X}_i$ is linearly correlated with $\mathbf{X}_j$. Because $\mathbf{X}_j$ is linearly correlated with $\mathbf{X}_k$ the matching process identifies $\mathbf{X}_k$ instead of $\mathbf{X}_j$ as a contributor - but that is not an issue as we know that $\mathbf{X}_k$ would also be linearly correlated with $\mathbf{X}_i$ and thus studying it will be just as useful.

For testing purposes however we will limit the sizes of the donor pool to try to minimise the probability of spurious contributions (as we have no ground truth data for confounding effects). Although this threatens the "sufficiently similar donors" assumption from Shi et al. [25] in crediting synthetic control as a viable method, we are no longer interested in this aspect of InterpTwin - section 5.1 shows us InterpTwin already performs soundly.

The testing procedure is outlined as follows for model $\mathcal{M}$ (all sampling is without replacement):

1. From a test dataset $(\mathbf{X}_i, \mathcal{T}_i^-, \mathbf{y}_i^-)_{i\in[N]}$ sample randomly $n_0 < N_0$ individuals from $\mathcal{I}_0$ and $n_1 < N_1$ from $\mathcal{I}_1$. Call these subsets I_0 and I_1. We used $n_0 = n_1$ in our experiments.
2. From the n_0 control individuals (WLOG, we can use treated individuals also) further sample 2-4 individuals that make up the true 'contributors'. Let these individuals be indexed by $i_1, i_2, \ldots, l_m$ for $m \in \{2, 3, 4\}$.
3. Randomly assign coefficients $a_1, \ldots, a_m$ a real value (within reason). If we do not allow extrapolation, then enforce non-negativity and adding up on the coefficients also. We create an *artificial* individual with covariates

$$\mathbf{X}_{\text{art}} = \sum_{i=j}^{m} a_j \mathbf{X}_{i_j}$$

4. Append $\mathbf{X}_{\text{art}}$ to the treated group I_1 which consists now of $n_1 + 1$ individuals. Use model $\mathcal{M}$ to obtain the estimated contributors from data $I_0 \cup I_1$.
5. Repeat from step 1 for L iterations.

We compare the estimated contributors against the true contributors using precision and recall metrics, choosing $L = 10^2$. Note in tables 2 and 3 only the *average* precision, recall and f1-scores are displayed (so the data shown does not necessarily follow f1= $pr/(p + r)$).

Table 2: Precision and recall scores for identifying the contributing set when **interpolating only**. Top two values of columns highlighted in bold. Empirical standard deviations for all entries are between 0.01-0.02.

Model		$n_0 = 10$			$n_0 = 20$			$n_0 = 40$		
Components	Lasso?	Pre	Rec	f1	Pre	Rec	f1	Pre	Rec	f1
SyncTwin	No	0.49	0.26	0.16	0.22	0.13	0.08	0.16	0.08	0.05
SyncTwin+$\mathcal{L}_s^*$	No	0.42	0.21	0.13	0.20	0.12	0.07	0.11	0.05	0.03
	Yes	0.53	0.45	0.23	0.35	0.31	0.16	**0.21**	**0.23**	**0.11**
SyncTwin+$\mathcal{L}_h$	No	0.45	0.24	0.15	0.22	0.11	0.07	0.14	0.08	0.05
	Yes	0.48	0.44	0.22	**0.37**	**0.33**	**0.17**	0.19	0.19	0.09
SyncTwin+$\mathcal{L}_s^*$+$\mathcal{L}_h$	No	0.41	0.28	0.16	0.22	0.11	0.07	0.10	0.07	0.04
	Yes	0.49	0.44	0.23	0.34	0.32	0.16	**0.20**	**0.23**	**0.11**
SyncTwin+$\mathcal{L}_s^{**}$	No	0.47	0.28	0.16	0.22	0.12	0.07	0.16	0.08	0.05
	Yes	**0.57**	**0.49**	**0.25**	**0.36**	**0.34**	**0.17**	0.17	0.18	0.09
InterpTwin	No	0.41	0.26	0.15	0.22	0.10	0.07	0.13	0.06	0.04
	Yes	**0.54**	**0.47**	**0.24**	0.31	**0.33**	0.15	0.19	0.18	0.09
Avg. Lasso diff		0.09	0.20	0.08	0.13	0.21	0.09	0.06	0.14	0.06

5.2.1 Discussion of results

Both in interpolation and extrapolation, our proposed methods have demonstrated capacity to outperform SyncTwin in all metrics (precision, recall, f1). Furthermore, these discrepancies increase with n_0, suggesting better robustness of our methods to spurious correlations.

The results within our ablated models don't give the most apparent trend as to showing the individual additions of $\mathcal{L}_s^*$, $\mathcal{L}_s^{**}$, $\mathcal{L}_h$ provide further validity. It is not significant out of all ablated versions, that InterpTwin produces the best results with interpolated data, although does show some marginal superiority when dealing with extrapolated data. Performing better on extrapolated individuals is expected, as we imagine inference within the convex hull of the covariates is already well trained on (thus less readily optimised) whereas the generalisation onto the extrapolated space requires some help from our regularisation.

Table 3: Precision and recall scores for identifying the contributing set when **allowing extrapolation**. Top two values of columns highlighted in bold. Empirical standard deviations for all entries are between 0.01-0.02.

Model		$n_0 = 10$			$n_0 = 20$			$n_0 = 40$		
Components	Lasso?	Pre	Rec	f1	Pre	Rec	f1	Pre	Rec	f1
SyncTwin	No	**0.52**	0.25	0.16	0.23	0.13	0.08	0.14	0.08	0.05
SyncTwin+$\mathcal{L}_s^*$	No	0.49	0.26	0.16	0.20	0.10	0.06	0.11	0.07	0.04
	Yes	0.48	0.37	0.20	**0.37**	**0.28**	**0.15**	0.20	0.16	0.08
SyncTwin+$\mathcal{L}_h$	No	0.30	0.15	0.09	0.27	0.12	0.08	0.14	0.08	0.05
	Yes	**0.50**	0.39	**0.21**	0.30	0.37	0.14	0.18	**0.17**	0.08
SyncTwin+$\mathcal{L}_s^*$+$\mathcal{L}_h$	No	0.43	0.23	0.14	0.24	0.12	0.08	0.16	0.08	0.05
	Yes	**0.50**	**0.40**	**0.21**	0.32	0.27	0.14	0.17	0.16	0.08
SyncTwin+$\mathcal{L}_s^{**}$	No	0.49	0.24	0.15	0.30	0.15	0.09	0.10	0.06	0.04
	Yes	0.48	0.36	0.19	0.31	0.27	0.14	**0.21**	**0.17**	**0.09**
InterpTwin	No	0.41	0.22	0.13	0.19	0.10	0.06	0.10	0.06	0.04
	Yes	**0.50**	**0.39**	**0.22**	**0.33**	**0.29**	**0.15**	**0.21**	**0.18**	**0.09**
Avg. Lasso diff		0.07	0.16	0.07	0.09	0.18	0.07	0.07	0.09	0.04

Table 4: Proportion of estimated contributors that reflectively select $\mathbf{X}_{\text{art}}$ as a contributor under both interpolation and extrapolation. Top percentages of columns highlighted in bold. Empirical standard deviations of all entries lie in-between 0.03-0.04.

Model	$n_0 = 10$		$n_0 = 20$		$n_0 = 40$		$n_0 = 80$		$n_0 = 160$	
	Int	Ext	Int	Ext	Int	Ext	Int	Ext	Int	Ext
SyncTwin+$\mathcal{L}_s^*$	**0.54**	**0.64**	0.39	**0.39**	0.13	0.26	0.08	0.16	0.09	0.16
SyncTwin+$\mathcal{L}_h$	0.43	0.57	0.36	0.47	0.25	**0.39**	**0.21**	**0.32**	0.12	0.23
SyncTwin+$\mathcal{L}_s^*$+$\mathcal{L}_h$	0.53	0.53	0.36	**0.51**	0.21	0.35	0.20	0.30	0.12	**0.25**
SyncTwin+$\mathcal{L}_s^{**}$	0.48	0.53	0.32	0.38	0.15	0.31	0.11	0.17	0.08	0.20
InterpTwin	0.47	0.61	0.32	0.49	**0.27**	0.37	0.18	0.26	**0.17**	0.23

The most interesting takeaway from validity testing is the hard-to-ignore performance gap between using/not using the Lasso method for matching latent vectors. In particular, the difference is magnified in recall scores as shown in the last rows of tables 2 and 3. The poor ability for SyncTwin's gradient-based method to recall the ground truth is most likely linked to hyperparameter choice of τ, the temperature for the Gumbel-Softmax function used by SyncTwin to approximately control the number of estimated contributors (similar effect as α in the Lasso). In SyncTwin, this parameter was chosen to optimise ITE error, and was particularly sensitive [20], so we left this untouched. This perhaps identifies a weakness of SyncTwin's matching method - the number of individuals chosen as contributors affects its ITE estimation significantly.

5.3 Consistency testing

In similar fashion to testing validity, we design a corresponding experiment for consistency - if i selected j as a contributor, does j select i? We focus only in comparisons of models that use Lasso, as section 5.2 already establishes its superiority in our proposed framework. The setup is similar:

1. Continue from after step 4 of the procedure outlined in 5.2. For each *estimated* contributor $\hat{i}_1, \ldots, \hat{i}_{\hat{m}}$, replace it with $\mathbf{X}_{\text{art}}$ in the control group I_0, and treat $\mathbf{X}_{\hat{i}_j}$ as part of the treatment group I_1.
2. Use $\mathcal{M}$ to estimate the contributors of $\mathbf{X}_{\hat{i}_j}$ in this augmented dataset. Repeat for $j = 1, \ldots, \hat{m}$ and record the proportion of cases that $\mathbf{X}_{\text{art}}$ was indeed in the estimated set.
3. Repeat from step 1 (in section 5.2) for L iterations.

5.3.1 Discussion of results

Table 4 reports results on consistency. We notice that across using either $\mathcal{L}_s^*$ or $\mathcal{L}_s^{**}$ (or neither) as supervised losses, there is no obvious trend to how it affects consistency values. Indeed this is quite expected (along with validity testing results too), as we were motivated to append these extra losses with the intention of fitting more robustly to the DGP assumption - which is not met in the case of our data generation [20]. In settings where DGP is falsely assumed, InterpTwin does not expect the addition of our proposed supervised losses to have any meaningful effect on results. For evidence that the losses $\mathcal{L}_s^*$ and $\mathcal{L}_s^*$ are indeed helpful, please see Appendix A.6 for further experimentation results.

Another important observation is the usefulness in adding a helper loss $\mathcal{L}_h$. Comparing SyncTwin+$\mathcal{L}_s^*$ with SyncTwin+$\mathcal{L}_s^*$+$\mathcal{L}_h$ and also SyncTwin+$\mathcal{L}_s^{**}$ with InterpTwin, we see the presence of a helper loss gives significant positive effects to consistency. This 'bonus' becomes more apparent with greater n_0 - which again suggests training with a helper loss gives more consistent predictions, robust to spurious correlations.

6 Conclusion and Future Work

InterpTwin has demonstrated its capabilities in (LIP) ITE estimation by preserving the ideas of its predecessor, SyncTwin. Moreover, we have shown on simulated data that InterpTwin has further strengths in producing more interpretable, robust results in synthetic weight calculation that may present more trustworthy information for the domain expert to use.

Given the nature of our 'interpretability' benchmarks in sections 5.2 and 5.3, it may be difficult to replicate such testing ideas with real data, but would most certainly still be worthy of future investigation. InterpTwin's performances (in all areas, ITE estimation included) on real observational datasets would be desired. Moreover, theoretical quantification of InterpTwin's linear regularisation under broader settings than that of Appendix A.4 ought to be explored. Although we anticipate such work would still require an assumptional framework, nevertheless presents as an interesting avenue of research.

References

[1] A. Abadie. Using synthetic controls: Feasibility, data requirements, and methodological aspects, June 2021. URL https://www.aeaweb.org/articles?id=10.1257/jel.20191450.

[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. doi: 10.1198/jasa.2009.ap08746. URL https://doi.org/10.1198/jasa.2009.ap08746.

[3] A. Abadie, A. Diamond, and J. Hainmueller. Comparative politics and the synthetic control method. *American Journal of Political Science*, 59(2):495–510, 2015. doi: https://doi.org/10.1111/ajps.12116. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/ajps.12116.

[4] J. H. Abbring and G. J. van den Berg. The nonparametric identification of treatment effects in duration models. *Econometrica*, 71(5):1491–1517, 2003. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1555509.

[5] T. Adel, Z. Ghahramani, and A. Weller. Discovering interpretable representations for both deep generative and discriminative models. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 50–59. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/adel18a.html.

[6] M. J. Amjad, D. Shah, and D. Shen. Robust synthetic control. *J. Mach. Learn. Res.*, 19:22:1–22:51, 2017.

[7] A. Andreella, R. D. Santis, A. Vesely, and L. Finos. Procrustes-based distances for exploring between-matrices similarity. *Statistical Methods & Applications*, 2023. URL https://api.semanticscholar.org/CorpusID:255941672.

[8] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL `https://api.semanticscholar.org/CorpusID:11212020`.

[9] F. Barbato, M. Toldo, U. Michieli, and P. Zanuttigh. Latent space regularization for unsupervised domain adaptation in semantic segmentation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2829–2839, 2021.

[10] J. S. Clark and O. N. Bjørnstad. Population time series: Process variability, observation errors, missing values, lags, and hidden states. *Ecology*, 85(11):3140–3150, 2004. ISSN 00129658, 19399170. URL `http://www.jstor.org/stable/3450552`.

[11] N. Doudchenko and G. Imbens. Balancing, regression, difference-in-differences and synthetic control methods: A synthesis. *ERN: Cross-Sectional Models*, 2016.

[12] G. Hadjeres, F. Nielsen, and F. Pachet. Glsr-vae: Geodesic latent space regularization for variational autoencoder architectures. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7, 2017.

[13] C. Heinze-Deml, S. Sippel, A. G. Pendergrass, F. Lehner, and N. Meinshausen. Latent linear adjustment autoencoder v1.0: a novel method for estimating and emulating dynamic precipitation at high resolution. *Geoscientific Model Development*, 14(8):4977–4999, 2021. doi: 10.5194/gmd-14-4977-2021. URL `https://gmd.copernicus.org/articles/14/4977/2021/`.

[14] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997. doi: 10.1162/neco.1997.9.8.1735.

[15] F.-X. Hong, X.-L. Zheng, and C.-C. Chen. Latent space regularization for recommender systems. *Information Sciences*, 360:202–216, 2016. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2016.04.042. URL `https://www.sciencedirect.com/science/article/pii/S002002551630295X`.

[16] Y.-N. Hung and A. Lerch. Feature-informed latent space regularization for music source separation. 2022.

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[18] S. MacDonald, K. Steven, and M. Trzaskowski. *Interpretable AI in Healthcare: Enhancing Fairness, Safety, and Trust*, pages 241–258. Springer Nature Singapore, Singapore, 2022. ISBN 978-981-19-1223-8. doi: 10.1007/978-981-19-1223-8_11. URL `https://doi.org/10.1007/978-981-19-1223-8_11`.

[19] B. M. Marlin, D. C. Kale, R. G. Khemani, and R. C. Wetzel. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, IHI '12, page 389–398, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450307819. doi: 10.1145/2110363.2110408. URL `https://doi.org/10.1145/2110363.2110408`.

[20] Z. Qian, Y. Zhang, I. Bica, A. Wood, and M. van der Schaar. Synctwin: Treatment effect estimation with longitudinal outcomes. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 3178–3190. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/19485224d128528da1602ca47383f078-Paper.pdf`.

[21] T. S. Richardson. Single world intervention graphs (swigs) : A unification of the counterfactual and graphical approaches to causality. 2013. URL `https://api.semanticscholar.org/CorpusID:126353329`.

[22] M. W. Robbins, J. M. Saunders, and B. Kilmer. A framework for synthetic control methods with high-dimensional, micro-level data: Evaluating a neighborhood-specific crime intervention. *Journal of the American Statistical Association*, 112:109 – 126, 2017.

[23] M. J. Schuemie, G. Hripcsak, P. B. Ryan, D. Madigan, and M. A. Suchard. Empirical confidence interval calibration for population-level effect estimation studies in observational healthcare data. *Proceedings of the National Academy of Sciences*, 115(11):2571–2577, 2018. doi: 10.1073/pnas.1708282114. URL `https://www.pnas.org/doi/abs/10.1073/pnas.1708282114`.

[24] M. Schulz and K. Stattegger. Spectrum: spectral analysis of unevenly spaced paleoclimatic time series. *Computers and Geosciences*, 23(9):929–945, 1997. ISSN 0098-3004. doi: https://doi.org/10.1016/S0098-3004(97)00087-3. URL `https://www.sciencedirect.com/science/article/pii/S0098300497000873`.

[25] C. Shi, D. Sridhar, V. Misra, and D. M. Blei. On the assumptions of synthetic control methods, 2021.

[26] R. L. Tannen, M. G. Weiner, and D. Xie. Use of primary care electronic medical record database in drug efficacy research on cardiovascular outcomes: comparison of database and randomised controlled trial findings. *BMJ*, 338, 2009. ISSN 0959-8138. doi: 10.1136/bmj.b81. URL `https://www.bmj.com/content/338/bmj.b81`.

[27] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996. ISSN 00359246. URL `http://www.jstor.org/stable/2346178`.

[28] H. Wong and D. B. Rubin. Rubin, d. b. (2005), "causal inference using potential outcomes: Design, modeling, decisions," "journal of the american statistical association," 100, 322-331: Comment by wong and reply. *The American Statistician*, 62(3):275–278, 2008. ISSN 00031305. URL `http://www.jstor.org/stable/27644049`.

[29] L. Zhang, Y. Wang, A. Ostropolets, J. J. Mulgrave, D. M. Blei, and G. Hripcsak. The medical deconfounder: Assessing treatment effects with electronic health records. In F. Doshi-Velez, J. Fackler, K. Jung, D. Kale, R. Ranganath, B. Wallace, and J. Wiens, editors, *Proceedings of the 4th Machine Learning for Healthcare Conference*, volume 106 of *Proceedings of Machine Learning Research*, pages 490–512. PMLR, 09–10 Aug 2019. URL `https://proceedings.mlr.press/v106/zhang19a.html`.

[30] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67(2):301–320, 2005. ISSN 13697412, 14679868. URL `http://www.jstor.org/stable/3647580`.

A Appendix

A.1 Confounding latent effects

By allowing estimated latent vectors $\tilde{\mathbf{c}}_i$ to depend on the pre-treatment outcomes $\mathbf{y}_i^-$, we can no longer ensure faithfulness to the DGP after training. This is because when training our model, we impose two things: (1) outcomes to depend linearly on $\tilde{\mathbf{c}}_i$ by DGP assumption and (2) $\tilde{\mathbf{c}}_i$ to be a latent representation of the covariates such that the reconstruction loss of the auto-encoder is minimised. Refer to these two criteria as the supervised and reconstruction losses respectively.

Letting $\mathbf{y}_i^-$ be part of the covariates $\mathbf{X}_i' = \begin{bmatrix}\mathbf{X}_i \ \mathbf{y}_i^-\end{bmatrix}$, a model's training state may reach a local minimum in a problematic parameter state of similar idea expressed in that of figure 3. Here for the sake of simplicity there is only one time dimension $t \in \mathcal{T} = \{0\}$ (the extension to more times is trivial), and latent sub-vector $\mathbf{v}_i \in \mathbb{R}^M$ is a direct latent representation of covariates $\mathbf{X}_i$ independent of $\mathbf{y}_i^-$. Since $\mathbf{q}$ is a free parameter during training, we may set it in this case to be $\mathbf{e}_1$. The end result is that we learn an auto-encoder that is degenerate in its dependency on its outcomes $\mathbf{y}_i^-$, yet still achieves perfect (and over-fitted) supervised loss and good reconstruction loss (particularly if M is large compared to the dimension of $\mathbf{y}_i^-$). Such circumstances may arise when the supervised loss is heavily weighted in importance and/or the DGP assumption is not truthful on the fitted data.

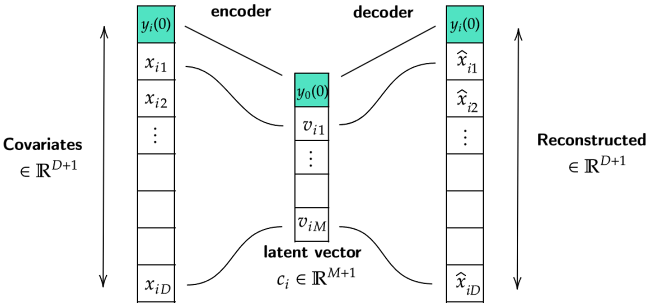

Figure 3: An example state of an auto-encoder which completely ignores structural assumptions from the DGP, yet may still lead to low training losses.

A.2 Synthetic control of latent space

The common assumption within synthetic control methods [1, 6, 25, 11] is that the outcomes $\mathbf{y}_i$ (more specifically, the *pre-treatment* outcomes) may be expressed as a linear combination of other donor outcomes for some donor set $\mathcal{D}$.

$$\mathbf{y}_i^-(0) = \sum_{j \in \mathcal{D}} \beta_j \mathbf{y}_j^-(0) \tag{9}$$

The notion that we may extend (9) to times $t \in \mathcal{T}^+$ is an important aspect that has been amiss in the literature, potentially implicitly believed. Nevertheless assuming this is true, under the DGP assumption and disregarding noise, we may re-write

$$\mathbf{Q}\mathbf{c}_i = \sum_{j \in \mathcal{D}} \beta_j \mathbf{Q}\mathbf{c}_j = \mathbf{Q} \sum_{j \in \mathcal{D}} \beta_j \mathbf{c}_j \tag{10}$$

where $\mathbf{Q} \in \mathbb{R}^{|\mathcal{T}| \times K}$ is the matrix with rows $\mathbf{q}_t$. So really the latent vectors need only to have linear representation in the (right) null space of $\mathbf{Q}$. The matching problem in (3) is therefore motivated by an assumption which is stronger than that of classical synthetic control, as in general $\mathbf{Q}$ does not have a left inverse (e.g. $K > |\mathcal{T}|$ or not full rank).

A.3 Training-independent latent vectors

We performed training of SyncTwin on the same generated dataset (section 5) $(\mathbf{X}_i, \mathcal{T}_i^-, \mathbf{y}_i^-)$ in two ways: once to predict the counterfactual control outcomes and another for the counterfactual treated outcomes. We extracted the latent factors of all N individuals in this dataset and concatenated them together to form two $K \times N$ matrices, corresponding to the two different training processes. Call these two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$.

In order to measure the discrepancies between the two versions of latent vectors of individuals, we upper bound the differences by considering the *Procrustes distance* [7] of $\mathbf{M}_1$ and $\mathbf{M}_2$:

$$d_P(\mathbf{M}_1, \mathbf{M}_2)^2 = \inf_{R \in U(N)} \left\| \bar{\mathbf{M}}_1 - R\bar{\mathbf{M}}_2 \right\|^2 \tag{11}$$

where $\bar{\mathbf{M}}_1, \bar{\mathbf{M}}_2$ are normalised versions such that $\operatorname{tr}(\bar{\mathbf{M}}_1 \bar{\mathbf{M}}_1^T) = 1$, $U(N)$ is the space of unitary matrix operators and the Frobenius matrix norm taken. Treating the latent space as a vector space (already optimistic), the Procrustes distance gives leeway for rotations, reflections and scaling differences.

We ran 20 training processes each for training with control/treated post-intervention outcomes with SyncTwin. We used $K = 40$ for the latent dimension, $N_0 = N_1 = 200$. All obtained pairwise Procrustes distances between the latent vectors of different training runs are shown in table 5. For better gauge of our metric value, we also generated random matrices drawn from Uniform($[-1, 1]^{K \times N}$) and compared it with the same matrix but with l random columns augmented with Gaussian noise of standard deviation σ.

Table 5: The average discrepancy (measured in d_P) between calculated latent vectors trained on asymmetric supervised loss. Blank $\mathbf{M}_2$ data means distances were calculated within $\mathbf{M}_1$ data.

Latent vector model		Procrustes distance ($\times 10^{-2}$)
$\mathbf{M}_1$ data	$\mathbf{M}_2$ data	
SyncTwin w/ $\mathcal{L}_s(\mathcal{D}_0)$	-	2.85 (0.12)
SyncTwin w/ $\mathcal{L}_s(\mathcal{D}_1)$	-	2.96 (0.09)
SyncTwin w/ $\mathcal{L}_s(\mathcal{D}_0)$	SyncTwin w/ $\mathcal{L}_s(\mathcal{D}_1)$	3.35 (0.08)
Uniform($[-1, 1]^{40 \times 400}$)	$l = 400$, $\sigma = 1$	3.65 (0.02)
Uniform($[-1, 1]^{40 \times 400}$)	$l = 400$, $\sigma = 0.6$	2.57 (0.01)
Uniform($[-1, 1]^{40 \times 400}$)	$l = 100$, $\sigma = 1$	3.36 (0.02)
Uniform($[-1, 1]^{40 \times 400}$)	$l = 5$, $\sigma = 2$	2.73 (0.04)

We see that the results in table 5 suggest that the variation between calculated latent vectors within different training runs under the *same* training loss is significantly less than the variation across different training losses. We can compare the difference to applying a standard unit Gaussian noise filter onto a quarter randomly chosen columns of a random uniform matrix, which is sizable amount of noise.

A.4 Linearity from discovering hidden confounders

For the sake of simplicity assume only one time dimension for our discussion. Suppose our list of covariates $\mathbf{X}_i \in \mathbb{R}^D$ does not include variable $\lambda_i \in \mathbb{R}$ which is a hidden confounder of $\mathbf{X}_i$ and $y_i \in \mathbb{R}$. From an information theoretic perspective, knowing λ_i gives us extra information about y_i and $\mathbf{X}_i$. Suppose λ_i for each i come from the same family of distributions (so have the same support), with marginals $p_i(\lambda)$, and have the following relationship with the covariates:

$$\mathbf{X}_i = f(\lambda_i) + \eta_i \tag{12}$$

where f is some function $f : \mathbb{R} \to \mathbb{R}^D$ and $\eta_i \in \mathbb{R}^D$ is a random vector which is independent to λ_i such that $\mathbb{E}(\eta_i) = 0$ for all i. For any vector of coefficients $\alpha = (\alpha_i)_{i \in \mathcal{D}}$, note

$$\begin{aligned}
\mathbb{E}(\mathbf{X}_i) - \sum_{k \in \mathcal{D}} \alpha_k \mathbb{E}(\mathbf{X}_k) &= \mathbb{E}(\mathbf{X}_i) - \sum_{k \in \mathcal{D}} \alpha_k \int_\lambda p_k(\lambda) \mathbb{E}(\mathbf{X}_k \mid \lambda) \, \mathrm{d}\lambda \\
&= \int_\lambda \left(p_i(\lambda) - \textstyle\sum_{k \in \mathcal{D}} \alpha_k p_k(\lambda) \right) \mathbb{E}(\mathbf{X}_i \mid \lambda) \, \mathrm{d}\lambda \\
&= \int_\lambda f(\lambda) \textstyle\sum_{k \in \mathcal{D}} \alpha_k \left(p_i(\lambda) - p_k(\lambda) \right) \mathrm{d}\lambda + (1 - \textstyle\sum_{k \in \mathcal{D}} \alpha_k) \mathbb{E}(\mathbf{X}_i) \quad (13)
\end{aligned}$$

Assume further that $\sum_{k \in \mathcal{D}} \alpha_k = 1$ (corresponds to adding up assumption), and that f is a bounded function in the sense that all its components f_j are bounded. Then we have the following proposition

Proposition 1. *Under the setting of previous discussion, we have*

$$\left\| \mathbb{E}(\boldsymbol{X}_i) - \sum_{k \in \mathcal{D}} \alpha_k \mathbb{E}(\boldsymbol{X}_k) \right\|_1 \leq 2 \cdot \left(\sum_{j=1}^{D} \|f_j\|_\infty \right) \cdot \left(\sum_{k \in \mathcal{D}} |\alpha_k| \, d_{TV}(p_i, p_k) \right) \tag{14}$$

where $d_{TV}(\cdot\,,\cdot)$ is the total variation distance of two probability measures.

Proof. Continued from (13) with our new assumptions, we have after applying the l_1 norm to both sides and using Hölder's inequality,

$$\begin{aligned} \left\|\mathbb{E}(\mathbf{X}_i) - \textstyle\sum_{k\in\mathcal{D}} \alpha_k \mathbb{E}(\mathbf{X}_k)\right\|_1 &\leq \sum_{j=1}^{D} \int_\lambda \left| f_j(\lambda) \textstyle\sum_{k\in\mathcal{D}} \alpha_k \left(p_i(\lambda) - p_k(\lambda)\right) \right| \mathrm{d}\lambda \\ \overset{\text{Hölder's}}{\Longrightarrow} \qquad &\leq \left(\textstyle\sum_{j=1}^{D} \|f_j\|_\infty\right) \cdot \int_\lambda \left|\textstyle\sum_{k\in\mathcal{D}} \alpha_k \left(p_i(\lambda) - p_k(\lambda)\right)\right| \mathrm{d}\lambda \\ &\leq \left(\textstyle\sum_{j=1}^{D} \|f_j\|_\infty\right) \cdot \left(\textstyle\sum_{k\in\mathcal{D}} |\alpha_k| \int_\lambda |p_i(\lambda) - p_k(\lambda)| \, \mathrm{d}\lambda\right) \\ &= 2 \cdot \left(\textstyle\sum_{j=1}^{D} \|f_j\|_\infty\right) \cdot \left(\textstyle\sum_{k\in\mathcal{D}} |\alpha_k| d_{\mathrm{TV}}(p_i, p_k)\right) \end{aligned}$$

□

Our interpretation of (14) is as follows. For individuals k truly 'similar' to i with regards to the hidden confounder λ, we expect $d_{\mathrm{TV}}(p_i, p_k)$ to be small. If set $\mathcal{D}$ is an accurate depiction of such 'similar' set, we expect the right hand side bound to be much smaller, and thus we are more likely to find an approximate linear relationship on the left hand side. We summarise:

"Set of individuals with some common trait with respect to a hidden confounder are likely to exhibit a linear relationship within their covariates."

Note the reverse implication is not true - a linear relationship amongst covariates does not imply a common hidden confounder. This is why further investigation under suitable domain expertise is required.

A.5 Encoding longitudinal data linearly

Treating data which varies along the time dimension as simply another covariate dimension may lead to over-fitting as it fails to capture the inherent temporal structure. SyncTwin (and InterpTwin also) approaches this with a standard attentive encoder [8] and a LSTM [14] decoder architecture.

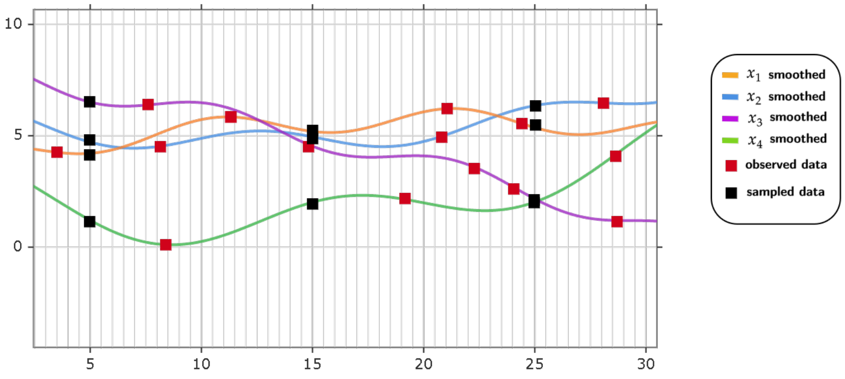

Figure 4: A make-do method for encoding temporal covariate structures in a linear manner.

For our helper loss, such methods do not satisfy the linear properties we desire. The implementation we used involves fitting our temporal covariate data with some relevant functional smoothing method (e.g. cubic splines), and then sampling new data points at uniform time intervals to produce a new covariate matrix in which we may take linear transformations of. This method unfortunately does not capture the temporal structure either, but does deal with irregular time recordings.

We are not satisfied with our make-do method described in figure 4, and identify it as an area requiring future work. However for the purposes of our results, along with the luxury of simulated data, what we used was enough.

A.6 Generating data under DGP

To test the validity of our 'improved' losses $\mathcal{L}_s^*$ and $\mathcal{L}_s^{**}$, we created a new artificial dataset which was generated strictly under our DGP assumption. The significance of such dataset has no real world resemblance, and was designed for the sole purpose of prompting differences in performance between our supervised losses.

Data Generation. For this data, let $D = K = 10$, $\mathcal{T} = \{i/30 : i = 1, \dots, 29\}$, $\mathcal{T}^- = \{i \in \mathcal{T} : i < 25/30\}$. Let $F \in \mathbb{R}^{K\times|\mathcal{T}|}$ be the matrix with entries $F_{ij} = f_i(t_j)$ where f_i is the ith Chebyshev polynomial and t_j is the jth time record and similarly with $F_1 \in \mathbb{R}^{K\times|\mathcal{T}^+|}$ for only the post-treatment times. Let $\Omega, \Omega^+ \in \mathbb{R}^{K\times K}$ be a random matrices with entries drawn from Uniform($[-1, 1]$). For control individuals, we define the matrix $\mathbf{Q}_C$ to be ΩF. For $\mathbf{Q}_T$ of treated individuals, we replace the last $|\mathcal{T}^+|$ columns of $\mathbf{Q}_C$ with $\Omega^+ F_1$. We generate for each i its covariates $\mathbf{X}_i \in \mathbb{R}^{D\times|\mathcal{T}^-|}$ with entries drawn from standard unit Gaussians. Our latent vectors and outcomes variables are related in the following way:

$$\mathbf{y}_i = \mathbf{Q}_C\mathbf{c}_i \text{ for control} \qquad \mathbf{y}_i = \mathbf{Q}_T\mathbf{c}_i \text{ for treated} \qquad \mathbf{c}_i = \tanh\left(\mathbf{X}_i\gamma\right) \tag{15}$$

for some random vector of coefficients $\gamma \in \mathbb{R}^{|\mathcal{T}^-|}$ again generated from Uniform($[-1, 1]$). On top of this, all covariates and outcomes are augmented with Gaussian noise of size σ.

Since the relationship between the latent vectors and covariates we generated is not too hard for neural nets to learn, we implemented the following techniques to exaggerate any difference in performances:

- Increase the noise levels. For these experiments we used $\sigma = 0.5$.
- Premature stopping of training. This way we may notice differences in how fast our models train with/without augmented losses. We trained on 500 iterations instead of the usual 5000.
- Reduce size of dataset, so models have less to work with. We used only $N_0 = N_1 = 10$ individuals per treatment group.
- Numerous training runs. Any differences will be small, but if repeated over many different starting random seeds, patterns (if significant) will emerge. We ran 200 trainings for each of $\mathcal{L}_s^*$, $\mathcal{L}_s^{**}$ and SyncTwin.

For these experiments we did not train with a helper loss and used the Lasso as our matching method. Under each new random seed training run, we noted which model had a lower MAE on the test set and referred our findings against a binomial distribution p-value test. Refer to table 6 for results.

Table 6: Comparing which ablated model version gave better MAE over 200 training runs.

First model	Second model	First lower	Second lower	p-value
SyncTwin + $\mathcal{L}_s$	SyncTwin + $\mathcal{L}_s^*$	83	117	0.010
SyncTwin + $\mathcal{L}_s$	SyncTwin + $\mathcal{L}_s^{**}$	87	113	0.038
SyncTwin + $\mathcal{L}_s^*$	SyncTwin + $\mathcal{L}_s^{**}$	86	114	0.028

Thus under a 5% significance level we have evidence to suggest that we have increasing performance $\mathcal{L}_s \prec \mathcal{L}_s^* \prec \mathcal{L}_s^{**}$ from our proposed losses.