A Novel Application of SCMs to Time Series Counterfactual Estimation in the Pharmaceutical Industry

Tomàs Garriga[*]
Novartis
Barcelona Supercomputing Center

Gerard Sanz[†]
Novartis

Eduard Serrahima de Cambra[‡]
Novartis

Axel Brando[§]
Barcelona Supercomputing Center

Abstract

In this paper, we present a novel application of structural causal models (SCMs) and the abduction-action-prediction procedure [1] to a time series setting in the context of a real world problem in the pharmaceutical industry. We aim to estimate counterfactuals for the sales volume of a drug that has been impacted by the entry to the market of a competitor generic drug. We employ encoder-decoder based architectures, applying a conditional variational autoencoder and also introducing the use of conditional sparse autoencoders, which had never been used in counterfactual literature. The proposed methodology requires availability of historical event and event-less time series and has the advantage of not relying on control covariates that may be unavailable, while clearly outperforming the basic counterfactual estimate of a forecast. We evaluate our approach using our company's real-world sales dataset, as well as synthetic and semi-synthetic datasets that mimic the problem context, demonstrating its effectiveness. We have successfully applied this model in our company, providing useful information for business planning, investment allocation and objectives setting.

1 Introduction

In many industries, knowing the impact of a *competitor entry* in sales is critical for business planning, investment allocation and objectives setting. The Pharma industry is notably affected when a drug's patent expires and Loss of Exclusivity (LOE) [2] takes place, prompting competitors to launch cheaper generic versions of the drug. This usually results in a dramatic decrease in the sales volume of about a 60-70% in the first years [2], severely affecting the company's revenues. Thus, accurately assessing the market impact of generic drug entries is of utmost importance.

Time series counterfactual estimation is an essential tool to understand an event's impact on time series data. It is applied to situations where an event or a treatment at a certain point in time alters a time series' trajectory, and consists in inferring, once given the observed post-event data, the counterfactual data, i.e., the time series that would have taken place if the event had not occurred. In the case of an event such as the entry in the market of a generic competitor drug, a straightforward

[*] tomas.garriga_dicuzzo@novartis.com | tomas.garriga@bsc.es
[†] gerard.sanz_estape@novartis.com
[‡] eduard.serrahima_de_cambra@novartis.com
[§] axel.brando@bsc.es

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

approach for counterfactual estimation is to simply use a time series forecasting model trained on historical data which was unaffected by this type of event, and predict the next steps of our pre-event time series according to the model. The main problem of this approach is that its estimations rely solely on pre-event data, lacking the ability to incorporate relevant information of the post-event facts that could have affected the counterfactual time series.

For example, let us imagine that few months after a generic drug enters into the market to compete with our target brand, an extreme weather event creates a problem in logistics which results in lower than expected sales in the whole pharmaceutical market. With a forecast based model we would estimate regular, not reduced, counterfactual sales volume despite the fact that, in all probability, our counterfactual time series would have been affected by the aforementioned happening. This would create a misconception of the impact that would be problematic for business analysis and planning.

The main method for time series counterfactual estimation that avoids this problem and captures information from post-event observations is synthetic control [3, 4]. Causal Impact [5], a powerful approach based on synthetic control principles, estimates counterfactuals with two data information sources: the pre-event part of the target time series, and control time series that were predictive of the target one prior to the intervention and have not been affected by it. This method, contrary to the previously exposed forecast based one, seizes information of post-event observations. The fact that it relies on time series other than the target one, however, can be a limitation in some cases. In our industrial context, for instance, the control data that can be accessed do not usually produce satisfactory results.

Based on our industrial application in the generics problem, in this paper we show that is possible to use SCMs and the abduction-action-prediction procedure [1] to infer time series counterfactuals, seizing information of post-event observations while not requiring access to any time series other than the target one at inference time, i.e., capturing the post-event information directly from the target time series. One limitation of our approach is that it requires availability of historical event and event-less time series. Previously, several works like [6, 7, 8, 9] have used SCMs to infer counterfactuals, but these methods have never been applied to time series settings. We use autoencoder based architectures for SCMs modelling and for the abduction step, and show that sparse autoencoders [10], not used previously in counterfactual literature, can perform better than the commonly used VAEs [11]. In appendix A, we provide an extensive analysis of the related work, both regarding synthetic control approaches for time series counterfactuals and SCMs for other types of counterfactuals.

This paper is structured as follows: Sec. 2 presents some background on SCMs. Sec. 3 details our methodology, covering description of our problem setting and the explanation of our approach. In Sec. 4, we introduce the datasets, models and metrics, and we discuss the results. Finally, Sec. 5 presents the conclusions of this work.

Our main contributions are: 1) a solution for an industrial problem in the pharmaceutical market based on SCMs and recent advances in causal machine learning, 2) an adaptation to a time series setting of the abduction-action-prediction procedure, always applied, until now, to other types of data, and 3) the use of sparse autoencoders for counterfactual estimation.

2 Background on Structural Causal Models

The proposed approach for counterfactual estimation is based on the J. Pearl definition of counterfactual [1], which can be operationalized by employing SCMs.

A SCM $\mathcal{M} := (\mathbf{S}, P(\boldsymbol{U}))$ consists of a collection $\mathbf{S} = \{f_i\}_{i=1}^N$ of structural assignments $a_i := f_i(u_i; \mathbf{pa}_i)$, where $\mathbf{pa}_i$ is the set of parents of a_i (its direct causes), and a joint distribution $P(\boldsymbol{U}) = \prod_{i=1}^N P(U_i)$ over mutually independent exogenous noise variables (i.e. unaccounted sources of variation) [6]. SCMs allow to perform counterfactual queries in a three-step procedure [1]: **1) Abduction:** predict the 'state of the world' (the exogenous noise $\boldsymbol{u}$) that is compatible with the observed data $\boldsymbol{a}$, i.e., infer $P_{\mathcal{M}}(\boldsymbol{U} \mid \boldsymbol{a})$; **2) Action:** perform an intervention (e.g. do($A_i := \widetilde{a}_i$)) to the counterfactual SCM which corresponds to the desired manipulation, which generates the modified counterfactual SCM $\widetilde{\mathcal{M}} := \mathcal{M}_{\mathbf{a}, do(\widetilde{a}_i)} = (\tilde{\mathbf{S}}, P_{\mathcal{M}}(\boldsymbol{U} \mid \boldsymbol{a}))$; **3) Prediction:** compute the quantity of interest based on the distribution entailed by the modified counterfactual SCM, $P_{\widetilde{\mathcal{M}}}(\boldsymbol{a})$.

3 Method

3.1 Problem Setting of our Industrial Application

Next, we define the structure of the time series setting over which our models perform counterfactuals. **Problem Statement.** Let $\boldsymbol{X} = \{\boldsymbol{x}^i\}_{i=1}^N$ be a set of observed time series $\boldsymbol{x}^i = (x_1^i, \dots, x_T^i)$ with T steps, with $T = T_1 + T_2$. Some of these time series are impacted by an event e^i at time step $T_1 + 1$ (the time series that are not impacted by the event are also assigned a value e^i that represents the lack of impact. Thus, we can divide the time series in a pre-event segment $\boldsymbol{h}^i = (h_1^i, \dots, h_{T_1}^i)$ with T_1 steps and a post-event segment $\boldsymbol{y}^i = (y_1^i, \dots, y_{T_2}^i)$ with T_2 steps. From now on, we will denote as $\boldsymbol{H}$, E and $\boldsymbol{Y}$ the variables referring, respectively, to the pre-event time series, the event and the post-event time series, and as $\boldsymbol{h}$, e and $\boldsymbol{y}$ to their specific realizations, often distinguishing among factual values (e_f and $\boldsymbol{y}_f$) and counterfactual values (e_{cf} and $\boldsymbol{y}_{cf}$). For an observation $\left\{\boldsymbol{h}^i, e_f^i, \boldsymbol{y}_f^i\right\}$, our objective is to estimate the counterfactual values $\boldsymbol{y}_{cf}^i$ in the hypothetical scenario where E had taken a different value ($E = e_{cf}^i$) while the rest of the factors of variation of $\boldsymbol{Y}$ were maintained.

Given the previously defined variables, we define a counterfactual function f such that $y_{cf} = \mathrm{f}(\boldsymbol{h}, e_f, \boldsymbol{y}_f, e_{cf})$. Then, our task is to find an approximated counterfactual function $\hat{\mathrm{f}}$ that, similarly to f, estimates $\hat{\boldsymbol{y}}_{cf}$.

In our company, we count on a large amount of historical sales volume data for many products and countries, a significant amount of which have been impacted by a generics entry to the market. Thus, from these data, we can build a time series dataset of a selected number of steps T which consists of non impacted time series, obtained by applying a T steps rolling window to the non impacted historical data, and impacted time series, obtained by taking, once selected a number of pre-event steps T_1 and post-event steps T_2 (with $T_1 + T_2 = T$), the windows that match this selection in the impacted historical time series.

In our counterfactual setting, E and $\boldsymbol{H}$ are the parents of $\boldsymbol{Y}$, and we consider datasets with two structures: 1) $\boldsymbol{H}$ being parent of E, becoming a confounder (confounded setting), and 2) no direct relation existing among $\boldsymbol{H}$ and E (unconfounded setting). On the other hand, $\boldsymbol{H}$ and E will be always intervened ($\boldsymbol{H}$ to its same factual value $\boldsymbol{h}$ and E to the counterfactual value e_{cf}). Therefore, only for the variable $\boldsymbol{Y}$ it will be necessary to estimate a structural assignment f_y and to abduct its exogenous noise U_y, responsible for its variation once given E and $\boldsymbol{H}$. Thus, it is not necessary to treat E and $\boldsymbol{H}$ as random variables generated by an exogenous noise susceptible to be abducted (and, in the confounded case, generated also by their parents). This makes counterfactual estimation in both the confounded and the unconfounded settings equivalent in a practical level. In our industrial application, where LOE occurs at a legally determined time unrelated to other factors that affect or are affected by the sales, out data can be considered unconfounded. However, we test our approach for both unconfounded and confounded synthetic and semi-synthetic datasets that imitate the main features of our market problem. Figure 1 shows the computational and the causal graph of our problem. For the models that we present in this work, we assume that our variables data generating process follow a causal graph like the one shown in figure 1, with no hidden confounders, and also positivity.

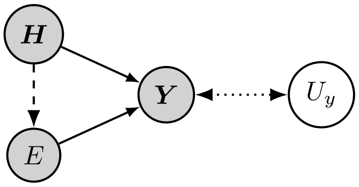

Figure 1: Problem setting's computational graph. Solid (and dashed) arrows depict influences on post-event time series $\boldsymbol{Y}$ in the unconfounded (confounded) setting, and dotted double arrows indicate a bidirectional relationship via the abduction step. Grey circles and solid (and dashed) arrows represent the causal graphs of the confounded (unconfounded) setting.

3.2 Encoder-Decoder Based Models for Counterfactual Estimation in our Problem Setting

Several approaches have been used for SCM modelling and counterfactual estimation for high dimensional settings. While some recent works use diffusion models [12], autoencoder based approaches are the most extended and effective. Next, we describe how our problem setting counterfactuals can be estimated using encoder-decoder architectures, assuming the graphs in figure 1.

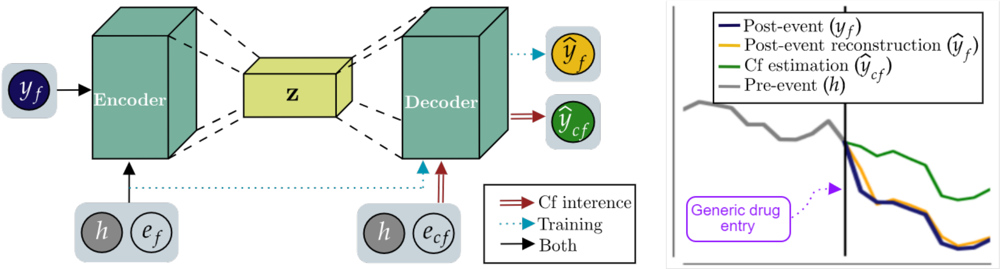

Figure 2: Scheme of encoder-decoder based methods for counterfactual inference, both in training and counterfactual inference phase.

Counterfactual estimation process. Considering the abduction-action-prediction procedure and our counterfactual problem setting, it is possible to define two trainable functions that, together, allow to estimate a counterfactual $\hat{\boldsymbol{y}}_{cf}$ once given $\boldsymbol{h}$, $\boldsymbol{y}_f$, e_f and e_{cf}. We will refer to the first of these functions as an encoder E_ϕ, with trainable parameters ϕ, which outputs either the location and scale parameters of a (usually multivariate) latent distribution $P(\boldsymbol{Z}|\boldsymbol{h}, e_f, \boldsymbol{y}_f) = E_\phi(\boldsymbol{h}, e_f, \boldsymbol{y}_f)$, or directly a latent variable $\boldsymbol{z} = E_\phi(\boldsymbol{h}, e_f, \boldsymbol{y}_f)$. $\boldsymbol{Z}$ is a latent representation that should allow the second of our trainable functions, which we call the decoder D_θ with trainable parameters θ, to estimate counterfactuals $\hat{\boldsymbol{y}}_{cf} = D_\theta(\boldsymbol{z}, \boldsymbol{h}, e_{cf})$. Figure 2 illustrates this process. In terms of the approximated counterfactual function mentioned in 3.1, we can express our functions as:

$$\hat{\boldsymbol{y}}_{cf} = \hat{\mathrm{f}}(\boldsymbol{h}, e_f, \boldsymbol{y}_f, e_{cf}) = D_\theta(E_\phi(\boldsymbol{h}, e_f, \boldsymbol{y}_f), \boldsymbol{h}, e_{cf}). \tag{1}$$

In the case where the encoder outputs the parameters of the latent distribution, the value $\boldsymbol{z}$ will be obtained via sampling from that distribution. There is a parallelism between this encoder-decoder setting and the abduction-action-prediction procedure. If we transfer the terminology of SCMs to our time series problem setting, we have that abduction step would consist in inferring $P_{\mathcal{M}}(U_y|\boldsymbol{h}, e_f, \boldsymbol{y}_f)$, which is analogue to our encoder estimating a conditional distribution of $\boldsymbol{Z}$ or directly its value. Here, $\boldsymbol{Z}$ is analogue to U_y, even if there can not be a complete identification among these variables [13]. Then, the decoder (that can be identified with the structural assignment f_y) inferring $\hat{\boldsymbol{y}}_{cf}$ would correspond simultaneously to the action and the prediction steps, where we first set the modified SCM $\widetilde{\mathcal{M}} = \mathcal{M}_{\mathbf{h}, \mathbf{e}_f, \mathbf{y}_f; do(E=e_{cf})}$ and then the result $\hat{\boldsymbol{y}}_{cf}$ is obtained as a sample of $\widetilde{\mathcal{M}}$.

Limitations of regular AEs. If we train a conditioned regular AE, like in figure 2, with only the reconstruction loss, and try to estimate counterfactuals, the results will not be satisfactory, as $\boldsymbol{Z}$ will account for all the factors of variation of $\boldsymbol{Y}$, and the conditioners will barely be used. To avoid that, some regularization technique over $\boldsymbol{Z}$ needs to be added to reduce the capacity of $\boldsymbol{Z}$ to seize information and force the model to use conditioners. Next, we explain the two encoder-decoder based models that we use to estimate counterfactuals.

Our models. First, we apply a conditional VAE (CVAE), with a KL [14] regularization term, to infer counterfactuals according to the previously defined process. Being CVAE a reasonable option to generate counterfactuals, some relevant problems exist. For example, the model can ignore, completely or partially, its conditioning $\boldsymbol{C}$. Some of the techniques that can be used to reduce the capacity of $\boldsymbol{Z}$, such as increasing the weight of KL or reducing the dimensionality of latent space, have severe consequences on the reconstruction capabilities, which also affect the quality of the counterfactuals estimations. Even without these techniques, the reconstruction capacity of VAEs is often poor, specially if we compare it to regular AEs [15]. For these reasons, we propose CSAE, a novel model for counterfactual estimation based on sparse autoencoders, that adds an L1 regularization term over $\boldsymbol{Z}$ to the regular reconstruction loss, limiting the capacity of $\boldsymbol{Z}$ to capture information, thus forcing the model to use conditionings. With this model, we maintain the reconstruction capacity of a regular autoencoder, adapting it to estimate sound counterfactuals. In appendix B, we provide a complete explanation of CVAE and CSAE.

4 Experimental Section

4.1 Datasets for Counterfactual Estimation

To assess the efficacy and validate the dependability of our techniques, we utilize multiple time series datasets exhibiting various characteristics. In counterfacual literature, there is the general problem of evaluation: as, in real world data, we do not have access to counterfactual ground truths, it is not possible to directly evaluate counterfactual models. Two alternatives exist to overcome this issue: the use of metrics that do not directly measure similarity among counterfactual estimations and ground truths but desirable properties of counterfactuals, which we address in 4.2, and the use of synthetic or semi-synthetic datasets, where we control the data generating process, or at least a part of it, and therefore have access to the counterfactual ground truths which allow to apply traditional metrics. Thus, we employ, apart from our property dataset, other datasets that imitate our industrial setting of generic competitor drugs entering into the market and causing severe changes in sales, but where counterfactual ground truths are known. All datasets are divided into time series with an event at the same given time step $(e = 1)$ and those without $(e = 0)$. Next, we describe them:

Real world dataset. This dataset shows the monthly sales of our company and has been built as explained in 3.1. We take 12 months as pre-event time steps and 30 months as post-event time steps. This is a private dataset as it contains company specific information.

Synthetic datasets. Dataset of time series with 30 steps that initiate always at value 0, have a trend that stems from a uniform distribution [-0.1,0.1] (meaning that this amount is added at each step), a drop of 0.7 in the step 20 in case of series with event, and an additional change at any randomly selected post-event step, with a value chosen uniformly within a range from -0.7 to 0.7. This value represents the effect of a happening that affects the time series both in case of event and without event, whereas the previous drop of 0.7 represents the effect of the main event. Besides, a Gaussian noise with variance of 0.1 is added to each step to make the time series more realistic. As can be noticed, the only difference in the generating process of these data for time series with and without event is the drop of 0.7, which allows to generate at the same time a time series with or without event and its counterfactual ground truth. There are two versions of this dataset: **1) Unconfounded**, where the presence or absence of event is completely random for every time series, and **2) Confounded**, where the probability of featuring an event follows a Bernouilli distribution with $p = (t + 0.1)/0.2$, there t is the trend.

Semi-synthetic dataset. This dataset is based in Rossmann Store Sales dataset [16], a public dataset from a Kaggle competition. It shows the daily sales of Rossmann drug stores from 2013 to 2015. We simulate a situation where, the first Monday of every march from 2013 to 2015, there is an event that affects half of the stores (e.g., it could be a promotion, a marketing campaign, etc.) and multiplies the sales by 1.1 the first day, 1.2 the second day, and 1.3 the rest of the days during three weeks. We use the four weeks previous to the first Monday of march of each year as pre-event time series, and the next three weeks as post-event time series. This is an unconfounded dataset.

Table 1: Results for time series datasets with both settings $e_f = 0$ and $e_f = 1$ over 10 random seeds. We measure MAE and MBE with respect to the counterfactual ground truth and the total and altered steps differences. Symbol $\sim$ means that the best results are those closest to the specified value.

Metric	Method	Synthetic 0	Synthetic 1	Conf. synth. 0	Conf. synth. 1	Semi-synth. 0	Semi-synth. 1	**Real world 0**	**Real world 1**
cf MAE $\downarrow$	LSTM	.199 ± .005	.198 ± .005	.199 ±.003	.201 ±.004	.101 ± .004	.080 ± .002	–	–
	CVAE	.144 ± .021	.137 ± .014	.145 ±.007	.187 ±.017	.105 ± .005	.083 ± .004	–	–
	CSAE	**.069 ± .004**	**.068 ± .009**	**.090 ± .030**	**.087 ± .012**	**.062 ± .004**	**.055 ± .003**	–	–
cf MBE ~ 0	LSTM	**.001 ± .011**	**.001 ± .014**	**-.001 ± .010**	**-0.016 ± .016**	.003 ± .004	**.002 ± .004**	–	–
	CVAE	−.067 ± .019	.067 ± .024	.051 ±.017	.126 ± .028	.011 ± .010	−.011 ± .009	–	–
	CSAE	.002 ± .008	.006 ± .016	.043 ± .048	-.041 ± .029	**-.001 ± .004**	**.002 ± .003**	–	–
Total Steps ~ 1	CVAE	.457 ± .091	.443 ± .066	.450 ±.038	.460 ± .040	.037 ± .011	.045 ± .110	**.899 ± .024**	1.372 ± .070
	CSAE	**.940 ± .037**	**..932 ± .069**	**.922 ± .045**	**.939 ± .056**	**.760 ± .046**	**.747 ± .042**	.845 ± .255	**1.192 ± .105**
Altered Steps ~ 1	CVAE	.388 ± .097	.360 ± .090	.241 ±.018	.246 ± .021	.109 ± .017	.109 ± .015	.312 ± .016	.710 ± .021
	CSAE	**.865 ± .009**	**.821 ± .059**	**.889 ± .043**	**.860 ± .029**	**.304 ± .015**	**.461 ± .009**	**.556 ± .213**	**.774 ± .094**

4.2 Evaluated Metrics

To evaluate our methods, we use various metrics. **Mean Absolute Error** (MAE) and **Mean Bias Error** (MBE) are employed to compare estimations with ground truths in synthetic and semi-synthetic datasets, as counterfactual ground truths exist solely for these datasets. MAE is a direct measure of how good our estimations are, while MBE allows to detect biases. We also consider additional metrics that evaluate some interesting properties of counterfactuals and do not require a ground truth, being then applicable to the real world dataset. Next, we present the Added Variations metrics.

Added Variations. We introduce this metric to evaluate how a counterfactual estimation responds to changes in the observations. The core idea is that, for an accurate method, in the case that we introduce variations in the post-event factual time series, the model should understand that they are the effect of some process that has nothing to do with the event and, therefore, should reflect those variations in the counterfactual estimate. This metric is implemented as follows: for each time series to be evaluated, several positive and negative values in the order of the data values are chosen; for each of this values, several windows of few consecutive steps from the post-event time series are selected and the chosen value is added to those steps. After that, a counterfactual estimate is obtained for every altered time series and it is compared to the counterfactual estimate of the non-altered time series. Two quantities are obtained: **1) Total difference**, that takes into account the difference among the altered counterfactual and the base counterfactual in all the steps, and **2) Altered steps difference**, which takes into account the difference among the altered counterfactual and the base counterfactual only in the steps affected by the alteration. These quantities are then divided by the expected difference (the product of the alteration value and the number of affected steps). Thus, ideally, the final results for both total difference and altered steps difference metrics should be 1. The final results are obtained by averaging all calculations. For a more formal definition, see Appendix C.

4.3 Models and Baselines

We compare the counterfactual estimations of CVAE and CSAE for all the metrics described in 4.2, and we add as a benchmark, for the MAE and MBE comparison with ground truth counterfactuals, an LSTM-based conditional forecast model that has as inputs only the pre-event time series and the value of the event. Thus, it can be used as a time series counterfactual estimator that does not take into account post-event values. Metrics like Added Variations only make sense for methods that use the abduction-prediction-action procedure.

Even if synthetic control methods would be a clear alternative to our approach, we exclude them from the evaluation because we are not working with control time series and, even if we could create synthetic datasets with control data to evaluate MAE, the results would depend much more on how predictive the control time series are than on the models themselves, making the comparison useless to evaluate method performances. This means that, in real applications, the selection of the optimal model should strongly depend on the control data, in case it exists. In our industrial context, existing control data has shown to not be of enough quality to raise the possibility of using these methods.

The encoder and decoder architectures of both CVAE and CSAE are shared, and are based on 1D convolutional and transposed convolutional layers, in a setting inspired in the VAE architecture for time series generation proposed in [17]. All methods have been implemented with TensorFlow [18], using an Adam optimizer with a learning rate of 10^{-4}. Other hyperparameters of CVAE and CSAE such as dimensionality of latent space or the factor of their respective regularizations (KL for CVAE and L1 for CSAE) are particular for each dataset and have been chosen after an optimization process.

4.4 Results

Table 1 shows the results of time series experiments for the datasets described in Sec. 4.1. Even if we are interested in counterfactuals for impacted time series, we evaluate the two possible settings: 0 when $e_f = 0$ and $e_{cf} = 1$, and 1 when $e_f = 1$ and $e_{cf} = 0$. All values have been obtained after performing 10 experiments with different random seeds, and the intervals correspond to the standard deviation. We see that, in counterfactual MAE metric, CSAE has the best results with an important difference. MBE metric allows to detect biases in the counterfactual estimations. For example, when using CVAE, the inferred counterfactual of time series are often biased towards the actual values. This is reflected in MBE metrics, where CSAE and LSTM model have similar values. In the Added Variations metrics, CSAE outperforms the other models in all settings except real world 0.

5 Conclusion

In this paper, we have adapted the theory of SCMs and the abduction-action-prediction procedure to the time series counterfactual of generic drug entry to the market. Two autoencoder based models have been proposed: CVAE, well known in counterfactual literature although not previously applied to time series problems, and CSAE, a new model for counterfactual inference. While CVAE shows some superiority with respect to the simple forecast counterfactual estimation in some cases, CSAE clearly outperforms both the forecast and CVAE, and makes the abduction-action-prediction process, under-explored in time series counterfactuals, an interesting option for cases similar to ours. The convenience of our approach over synthetic control techniques will depend on the availability and informative capacity of control data and on the amount of historical event and event-less time series. In our industrial context, CSAE has been applied to estimate the market impacts of generic entries, becoming a useful tool for the business planning area.

6 Acknowledgements

We gratefully acknowledge Novartis for sponsoring the industrial PhD. We also thank the Government of Catalonia's Industrial PhDs Plan for funding part of this research. We ackowledge Horizon Europe Programme under the AI4DEBUNK Project (`https://www.ai4debunk.eu`), grant agreement num. 101135757. Additionally, this work has been partially supported by PID2019-107255GB-C21 funded by MCIN/AEI/10.13039/501100011033 and JDC2022-050313-I funded by MCIN/AEI/10.13039/501100011033al by European Union NextGenerationEU/PRTR.

References

[1] Judea Pearl. Causality. *New York: Cambridge University Press*, 2000.

[2] Micael Castanheira, Carmine Ornaghi, and Georges Siotis. The unexpected consequences of generic entry. *Journal of Health Economics*, 68:102243, 2019.

[3] Janet Bouttell, Peter Craig, James Lewsey, Mark Robinson, and Frank Popham. Synthetic control methodology as a tool for evaluating population-level health interventions. *J Epidemiol Community Health*, 72(8):673–678, 2018.

[4] 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.

[5] Kay H. Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L. Scott. Inferring causal impact using bayesian structural time-series models. *Annals of Applied Statistics*, 9:247–274, 2015.

[6] Nick Pawlowski, Daniel Coelho de Castro, and Ben Glocker. Deep structural causal models for tractable counterfactual inference. *Advances in Neural Information Processing Systems*, 33:857–869, 2020.

[7] Álvaro Parafita and Jordi Vitrià. Explaining visual models by causal attribution, 2019.

[8] Álvaro Parafita and Jordi Vitrià. Estimand-agnostic causal query estimation with deep causal graphs. *IEEE Access*, 10:71370–71386, 2022.

[9] Pablo Sanchez-Martin, Miriam Rateike, and Isabel Valera. Vaca: Design of variational graph autoencoders for interventional and counterfactual queries. *arXiv preprint arXiv:2110.14690*, 2021.

[10] Andrew Ng et al. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19, 2011.

[11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2022.

[12] Pedro Sanchez and Sotirios A. Tsaftaris. Diffusion causal models for counterfactual estimation. *arXiv preprint arXiv:2202.10166*, 2022.

[13] Miguel Monteiro, Fabio De Sousa Ribeiro, Nick Pawlowski, Daniel C. Castro, and Ben Glocker. Measuring axiomatic soundness of counterfactual image models. *arXiv preprint arXiv:2303.01274*, 2023.

[14] Peter Hall. On kullback-leibler loss and density estimation. *The Annals of Statistics*, pages 1491–1519, 1987.

[15] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Towards deeper understanding of variational autoencoding models, 2017.

[16] Will Cukierski FlorianKnauer. Rossmann store sales, 2015.

[17] Abhyuday Desai, Cynthia Freeman, Zuhui Wang, and Ian Beaver. Timevae: A variational auto-encoder for multivariate time series generation. *arXiv preprint arXiv:2111.08095*, 2021.

[18] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[19] Brantly Callaway, Andrew Goodman-Bacon, and Pedro HC Sant'Anna. Difference-in-differences with a continuous treatment. Technical report, National Bureau of Economic Research, 2024.

[20] Susan Athey, Mohsen Bayati, Nikolay Doudchenko, Guido Imbens, and Khashayar Khosravi. Matrix completion methods for causal panel data models. *Journal of the American Statistical Association*, 116(536):1716–1730, 2021.

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

[22] Guido W Imbens and Donald B Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge university press, 2015.

[23] Sana Tonekaboni, Shalmali Joshi, David Duvenaud, and Anna Goldenberg. Explaining time series by counterfactuals, 2020.

[24] Licheng Liu, Ye Wang, and Yiqing Xu. A practical guide to counterfactual estimators for causal inference with time-series cross-sectional data. *American Journal of Political Science*, 2022.

[25] 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*, pages 884–895. PMLR, 2020.

[26] Ioana Bica, Ahmed M. Alaa, James Jordon, and Mihaela van der Schaar. Estimating counterfactual treatment outcomes over time through adversarially balanced representations, 2020.

[27] Wasim Ahmad, Maha Shadaydeh, and Joachim Denzler. Causal inference in non-linear time-series using deep networks and knockoff counterfactuals. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 449–454. IEEE, 2021.

[28] Valentyn Melnychuk, Dennis Frauen, and Stefan Feuerriegel. Causal transformer for estimating counterfactual outcomes, 2022.

[29] Carlos Carvalho, Ricardo Masini, and Marcelo Medeiros. Arco: An artificial counterfactual approach for high-dimensional panel time-series data. *Journal of Econometrics*, 207, 09 2018.

[30] Ivan Kobyzev, Simon JD Prince, and Marcus A Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE transactions on pattern analysis and machine intelligence*, 43(11):3964–3979, 2020.

[31] Saloni Dash, Vineeth N Balasubramanian, and Amit Sharma. Evaluating and mitigating bias in image classifiers: A causal perspective using counterfactuals. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 915–924, 2022.

[32] Xinwei Shen, Furui Liu, Hanze Dong, Qing LIAN, Zhitang Chen, and Tong Zhang. Disentangled generative causal representation learning, 2021.

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

[34] Guillaume Jeanneret, Loïc Simon, and Frédéric Jurie. Diffusion models for counterfactual explanations. In *Proceedings of the Asian Conference on Computer Vision*, pages 858–876, 2022.

[35] Anwaar Ulhaq, Naveed Akhtar, and Ganna Pogrebna. Efficient diffusion models for vision: A survey. *arXiv preprint arXiv:2210.09292*, 2022.

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

[37] Arnaud Van Looveren and Janis Klaise. Interpretable counterfactual explanations guided by prototypes. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 650–665. Springer, 2021.

[38] Giambattista Parascandolo, Mateo Rojas-Carulla, Niki Kilbertus, and Bernhard Schölkopf. Learning independent causal mechanisms. *CoRR*, abs/1712.00961, 2017.

[39] Murat Kocaoglu, Christopher Snyder, Alexandros G. Dimakis, and Sriram Vishwanath. Causalgan: Learning causal implicit generative models with adversarial training. *arXiv preprint arXiv:1709.02023*, 2017.

[40] Rui Liu, Yu Liu, Xinyu Gong, Xiaogang Wang, and Hongsheng Li. Conditional adversarial generative flow for controllable image synthesis. *CoRR*, abs/1904.01782, 2019.

[41] Mengyue Yang, Furui Liu, Zhitang Chen, Xinwei Shen, Jianye Hao, and Jun Wang. Causalvae: Structured causal disentanglement in variational autoencoder. *arXiv preprint arXiv:2004.08697*, 2022.

[42] Amar Kumar, Nima Fathi, Raghav Mehta, Brennan Nichyporuk, Jean-Pierre R Falet, Sotirios Tsaftaris, and Tal Arbel. Debiasing counterfactuals in the presence of spurious correlations. In *Workshop on Clinical Image-Based Procedures*, pages 276–286. Springer, 2023.

[43] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. GANITE: Estimation of individualized treatment effects using generative adversarial nets. In *International Conference on Learning Representations*, 2018.

[44] Athanasios Vlontzos, Bernhard Kainz, and Ciarán M. Gilligan-Lee. Estimating the probabilities of causation via deep monotonic twin networks. *CoRR*, abs/2109.01904, 2021.

[45] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[46] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.

A Extended Related Work

A.1 Time Series Counterfactuals

Several methods have been proposed for time series counterfactual estimation and causal impact inference in the presence of an event or treatment. Difference in differences [19] is a common approach which requires some control time series that are not affected by the event to estimate counterfactuals. This method features some important limitations like the Parallel Trends Assumption. Synthetic control [3, 4], which generalizes Difference in differences and overcomes some of its limitations, selects several available time series, other than the target one, that have been not affected by the event, computes some weights based on their pre-event similarity to the target time series, and then estimates the counterfactual as the weighted average of the post-event control time series. Causal Impact [5] is closely related to the synthetic control approach, and its main difference is that it estimates counterfactuals through a model that predicts the target from control series trained with pre-event observations. Matrix Completion Methods [20] are an alternative that can be viewed as a combination of synthetic control and the more classical unconfoundedness approach [21, 22].

Apart from the previous methods, there are other works which perform time series counterfactuals and have some relation with ours. [23] proposes a method for time series explanation based on counterfactuals. However, their authors use forecast methods that do not take into account actual observations and perform something more similar to an intervention. [24] imputes counterfactual outcomes for treated observations to estimate the average treatment effects (ATEs) using techniques like fixed effects counterfactual estimator, interactive fixed effects counterfactual estimator or matrix completion estimator. [25] develops a method that leverages the assignment of multiple treatments over time to estimate treatment effects in the presence of multi-cause hidden counfounders. [26] uses adversarially balanced representations to estimate counterfactuals of treatments at random time steps, which differs from our setting as we do not need to model random time steps events. [27] proposes a method to detect causal relationships in time series. [28] also proposes a method for counterfactual estimation with time varying treatments and counfounders based in transformers. ARCO [29] uses a combination of machine learning and time series econometrics techniques to estimate the causal effects of a treatment on an outcome variable in a panel time series data setting when a single unit is treated and control group is not available.

A.2 Causal Deep Leaning

In recent years, many works have appeared that mix the structural causal model (SCM) theory [1] and deep learning techniques to estimate counterfactuals. Apart from the ones mentioned in the paper, based on VAEs [11] and, to a lesser extend, normalizing flows [30], other interesting approaches have been proposed. For example, [31] and [32] use GANs [33], while [34] uses diffusion models [35]. [13] presents some useful metrics to evaluate counterfactuals, which are used in our paper to evaluate the models. [36] uses deep neural networks to disentangle object shape, object texture and background in natural images. [37] utilizes class prototypes in order to find interpretable counterfactual explanations. [38] uses multiple competing models in order to retrieve a set of independent mechanisms from a set of transformed data points in an unsupervised way.

There are other interesting works about causal representation learning with deep generative models that do not tackle directly the problem of counterfactual inference but are interesting to take into account. [39] and [40] combine GANs [33] with SCMs, basing a generator architecture on an assumed causal graph. However, this method lack tractable abduction capabilities and therefore cannot generate counterfactuals, reaching only the second rung of the causal ladder [1]. [41] proposes a method that learns a causal model, including the directed acyclic graph (DAG), over latent variables from data, and generates counterfactual samples. [42] uses a GAN based approach to address the specific problem of spurious correlations in medical datasets.

Finally, there are other interesting counterfactual estimation approaches that are applied to tabular data. Among them, [43], based in GANs, and [44], based in Deep Twin Networks, similar to siamese networks [45], stand out.

B Extended Models Explanation

The loss function for a classic CVAE [11] with a β penalty [46], which corresponds to the evidence lower bound (ELBO), for a datum x and a conditioning $\mathbf{c}$, is given by:

$$\mathcal{L}_{\text{CVAE}}(\theta, \phi) = \mathbb{E}_{q_\phi(\boldsymbol{z}|x,\mathbf{c})}[\log p_\theta(x|\boldsymbol{z}, \mathbf{c})] - \beta\, KL[q_\phi(\boldsymbol{z}|x, \mathbf{c}) \parallel p(\boldsymbol{z}))] \tag{2}$$

where both $q_\phi(\boldsymbol{z}|x, \mathbf{c})$ and $p_\theta(x|\boldsymbol{z}, \mathbf{c})$ are a set of dimension-wise independent normal distributions parameterised, respectively, by an encoder neural network E_ϕ and a decoder neural network D_θ, $p(\boldsymbol{z})$ is an isotropic normal prior distribution, KL is the Kullback–Leibler divergence [14] and β is a penalization over KL [46]. In the model training, the ELBO function is maximized with respect to parameters of the neural networks using the re-parametrization trick to sample from the approximate latent posterior: $\boldsymbol{z} = \mu_\phi(x, \mathbf{c}) + \alpha_\phi(x, \mathbf{c}) \odot \epsilon_{\boldsymbol{z}} \sim \mathcal{N}(0, I)$.

Based on the time series setting and the encoder-decoder counterfactual estimation method described in the main paper, it is possible to generate a counterfactual sample $\hat{\boldsymbol{y}}_{cf}$ by encoding an observation $\hat{\boldsymbol{y}}_f$ and its parents $\boldsymbol{h}$ and e_f, i.e. obtaining the normal distribution $q_\phi(\boldsymbol{z}|\boldsymbol{y}_f, \boldsymbol{h}, e_f)$, where position and scale parameters come from the encoder: $\mu_\phi, \alpha_\phi = E_\phi(\boldsymbol{y}_f, \boldsymbol{h}, e_f)$, then sampling the latent posterior from this distribution: $\boldsymbol{z} \sim q_\phi(\boldsymbol{z}|\boldsymbol{y}_f, \boldsymbol{h}, e_f)$, and finally decoding it along with the counterfactual event e_{cf}: $\hat{\boldsymbol{y}}_{cf} \sim p_\theta(\boldsymbol{y}_f|\boldsymbol{z}, \boldsymbol{h}, e_{cf})$. Notice that, in a practical level, the counterfactual will be decoded from the latent sample in a deterministic way: $\hat{\boldsymbol{y}}_{cf} = D_\theta(\boldsymbol{z}, \boldsymbol{h}, e_{cf})$.

Our proposed model, the Conditional Sparse Autoencoder (CSAE), is trained with the following loss function:

$$\mathcal{L}_{\text{CSAE}}(\mathbf{x}) = |x - \hat{x}| - \lambda \sum_i |z_i| \tag{3}$$

where x is the input variable, z_i are the elements of the latent space vector $\mathbf{z} = E_\phi(x, \mathbf{c})$ with $\mathbf{c}$ being the conditioning, λ is the hyperparameter of the penalty term, which in this work we have chosen to be L1, and $\hat{x}$ is the reconstruction term which stems from the decoder: $\hat{x} = D_\theta(z, \mathbf{c})$.

As in the case of CVAE, the framework described in 3.2 allows to generate a counterfactual sample $\hat{\boldsymbol{y}}_{cf}$ by encoding an observation $\hat{\boldsymbol{y}}_f$ and its parents $\boldsymbol{h}$ and e_f. In this case, we have $\mathbf{z} = E_\phi(\boldsymbol{y}_f, \boldsymbol{h}, e_f)$, and then $\hat{\boldsymbol{y}}_{cf} = D_\theta(\mathbf{z}, \boldsymbol{h}, e_{cf})$. Also similarly to CVAE, there is a parallelism among the abduction-action-prediction scheme and how CSAE performs counterfactuals; thus, z would make the function of the abducted exogenous noise and the action refers to the introduction of the counterfactual parents in the decoder instead of the factual ones.

C Added Variations Equations

Let $\boldsymbol{y} = \{\boldsymbol{y}_t\}, t \in T$ be the post-event time series over which we want to perform counterfactuals, $\boldsymbol{h}$ its correspondent historical time series previous to the event, e_f the (factual) event, and $\hat{\boldsymbol{y}}_{cf} = \hat{\text{f}}(\boldsymbol{y}, \boldsymbol{h}, e_f, e_{cf})$, where $\hat{\text{f}}$ is a counterfactual function and e_{cf} is the counterfactual event, its correspondent counterfactual estimation. Then, we consider a time series $A = 0...0, v_A...v_A, 0...0$ with T steps, where v_A is the value of the alteration which is added only to a certain number of consecutive steps. Let $\boldsymbol{y}^A = \boldsymbol{y} + A$ be the altered time series, then $\hat{\boldsymbol{y}}_{cf}^A$ would be its correspondent counterfactual estimation. We consider that, if our counterfactual model is correct, alterations in the factual time series should be reflected in the counterfactual time series. Thus, ideally $\sum_i \hat{\boldsymbol{y}}_{cf(i)}^A - \hat{\boldsymbol{y}}_{cf(i)} = \sum_i A_i = n_A \cdot v_A$, where n_A is the number of steps affected by the alteration in A. Taking into account that we use different time series A with different values n_A and v_A, we can express total differences metric for a single time series $\boldsymbol{y}$ (TD) as:

$$TD = \left\langle \frac{\sum_i \hat{\boldsymbol{y}}_{cf(i)}^A - \hat{\boldsymbol{y}}_{cf(i)}}{n_a \cdot v_A} \right\rangle_A, \tag{4}$$

and altered step differences (ASD) as

$$ASD = \left\langle \frac{\sum_i \hat{\boldsymbol{y}}_{cf(i)}^A - \hat{\boldsymbol{y}}_{cf(i)}}{n_a \cdot v_A} \mathbb{I}_{i \in s_A} \right\rangle_A, \tag{5}$$

where s_A is the set of altered steps (those with value v_A and not 0) in A and $\mathbb{I}$ is the indicator function. We see that, ideally, the result of these averages over the different alteration schemes should be 1. The results given in the paper are the averages of these metrics over all the time series in the test set. The parameters n_A, s_A and v_A are particular for every dataset and can be seen in the code.