Empirical Evaluation of the Performance of CEVAE under Misspecification of the Latent Dimensionality

Patrik Barták
Supervisor(s): Stephan Bongers, Jesse Krijthe
EEMCS, Delft University of Technology, The Netherlands

June 19, 2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

Causal machine learning deals with the inference of causal relationships between variables in observational datasets. For certain datasets, it is correct to assume a causal graph where information about unobserved confounders can only be obtained through noisy proxies, and CEVAE aims to address this case. The number of dimensions of the latent space modelled by CEVAE must be specified ahead of time, and this paper investigates the effect of this dimensionality misspecification on the performance of CEVAE. Results support the idea that underspecification and overspecification both degrade the performance of CEVAE, but indicate that underspecification is worse, at least for the case with few confounders. In general, the model does not always achieve best performance when the model dimensionality corresponds to the data dimensionality. Finally, conclusions made on data with linear-Gaussian proxies are the same as those obtained with nonlinear-Gaussian proxies, which indicates these conclusions generalize over different datasets to some extent.

1 Introduction

According to Guo et al. (2018), causality is a foundational AI problem, and has also been studied in a variety of high impact domains such as medical science, economics, epidemiology, and more. Louizos et al. (2017) also state that knowledge of causal effects is an area of increasing importance to policy makers. Examples of applications of the field include inferring the effectiveness of medication for different individuals, or examining the impact of regulations, policies, and other systematic interventions.

Causal effects can be studied from two main perspectives: causal discovery, and causal inference Guo et al. (2018). These respectively consist of: determining whether causal relationships exist between variables, and determining how manipulating a specific *treatment* variable would influence the value of a specific *outcome* variable. Causal inference, which is the focus of this research, in other words studies the extent to which variables are related through causation, rather than association.

Inferring causal effects using machine learning has been done using a variety of approaches, and these are referred to as *causal* machine learning. Guo et al. (2018) surveys several of these methods, which are briefly highlighted below.

Propensity Score Matching (Rosenbaum and Rubin, 1983) is an early method used to subdivide data into groups with similar treatment propensities (probabilities of an instance having some treatment, given some set of covariates). The authors show that this makes the distributions of treated and untreated subjects similar, reducing confounding bias.

Recent work uses neural networks to infer the counterfactual outcomes directly according to Guo et al. (2018), by fitting a function $f(\mathbf{x}, t)$ on factual outcome y. Here, $\mathbf{x}$ are the covariates and t is the treatment. Natural language processing techniques Pham and Shen (2017), TarNET (Shalit et al., 2016), and Balancing Counterfactual Regression (Johansson et al., 2016) have advanced this concept by learning low-dimensional representations $\mathbf{h}$ of the covariates $\mathbf{x}$, and using these for training and inference.

The Causal Effect Variational Autoencoder (CEVAE) (Louizos et al., 2017) is another step in the direction of representation learning, although from a different perspective. CEVAE claims to infer causal effects in cases with unobserved confounding variables. This is significant because there exist confounding variables that are either difficult, or impossible, to measure directly. Instead, CEVAE uses proxy variables that are affected by these latent, unmeasured confounders, but that do not themselves affect the treatment or the outcome. An example would be using mental health and wealth as proxies to a person's quality of life. Treating these proxies as the latent variables leads to bias, if they are not the true confounders, which is shown by the authors. Therefore, approaches such as CEVAE instead estimate a representation of the latent variable distribution from information embedded in the observational data.

The authors of CEVAE claim that an advantage of the model is that it makes weaker assumptions about the representation of the latent variable (whether it is continuous, discrete, or multidimensional) compared to analytical methods such as those introduced by Pearl (2012) and Kuroki and Pearl (2014). Still, the architecture of the VAE-based model requires that a choice is made regarding the model's latent distribution and the number of latent dimensions. To motivate their claim, the authors therefore perform the following synthetic experiment. They fit data from a data generating process with a Bernoulli latent variable to two CEVAE models: one with a 1-dimensional Bernoulli latent distribution, and one with a 5-dimensional normal latent distribution. Results show that both models can model the latent space, but the misspecified model requires a larger data sample to converge to a similar estimate.

In principle, this claim is also backed up by Kingma and Welling (2019), who claim that a multivariate normal distribution, in combination with the decoder of a VAE, can model complex distributions while remaining computationally tractable.

Still, the experimental evidence itself is limited. It does not address the issue of dimensionality in detail, and is performed on a relatively simple data generating process. The authors portray CEVAE as "tailored to the surrogate-rich setting when many proxies are available" (Louizos et al., 2017, p. 2), yet the experiment uses only one proxy variable to model a single latent variable. In a set of separate experiments, Rissanen and Marttinen (2021) provide counterexamples showing that CEVAE does not estimate causal effects correctly under model misspecification in the general case.

The question therefore remains: how do these conditions affect the performance of CEVAE? The focus is on dimensionality in order to limit the scope of this research, so a detailed look at distribution misspecification is left for future work. Addressing this issue is relevant because the actual latent dimensionality of proxies in datasets is rarely known. To specify, the following research question is addressed:

1. What is the impact of misspecified dimensionality on the performance of CEVAE?

These questions specify the scope of the research, and will be directly answered by this paper. First, a more detailed background on causal effect estimation and the CEVAE model is provided in Section 2. The methodology designed to answer the research questions is explained in Section 3. This is followed by the details of the experimental setup in Section 4, results of the experiments in Section 5, a detailed discussion of the results in Section 6, conclusions in Section 7, proposals for future work in Section 8, and an overview of responsible research in Section 9.

2 Background

The following section provides prerequisite background information. In particular, Section 2.1 explains relevant terms and concepts from causal inference literature, and introduces relevant notation. Section 2.2 provides a high-level overview of the CEVAE model.

2.1 Causal Effect Estimation

The goal of causal inference is to determine the causal effect of some variable t on another variable y (often called the treatment, and outcome, respectively) in an observational dataset. When referring to a specific data instance i, these will be denoted with a subscript (e.g: y_i). Often a set of covariates, $\mathbf{x}$, will have a causal effect on both t and y. In medical data, this could be for example age, as it will influence both the probability of an individual taking some medication, and the outcome of some illness. When attempting to find out if the medication changes the outcome of the illness, a spurious association will be seen even if no causal effect exists between t and y. These covariates are then called confounding variables, and they create bias in the estimate called confounding bias. For more detail on confounding bias, Guo et al. (2018) provide a comprehensive explanation.

Using the framework of (Pearl, 2009), namely Structural Causal Models (SCM) and do-calculus, the individualized treatment effect (ITE) for an instance with covariates $\mathbf{x}_i$ can be calculated:

$$\text{ITE} : \tau(\mathbf{x}_i) = \mathbb{E}[y|\mathbf{x} = \mathbf{x}_i, do(t = 1)] - \mathbb{E}[y|\mathbf{x} = \mathbf{x}_i, do(t = 0)] \tag{1}$$

The treatment does not need to be binary, but this is a common case ($t = 1$ for treatment and $t = 0$ control), and will be assumed in order to narrow the scope of the research.

It is also possible to define the average treatment effect (ATE), which describes the effect of the intervention on the entire population, rather than on a specific instance:

$$\text{ATE} : \mathbb{E}[\tau(\mathbf{x})] \tag{2}$$

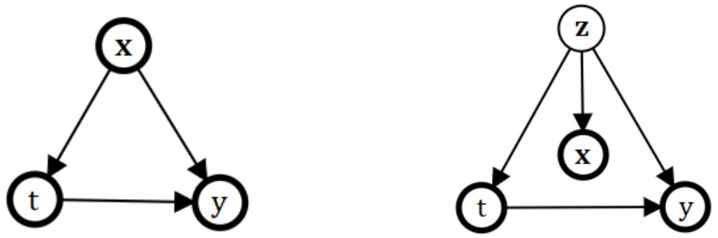
It is the goal of causal machine learning models to estimate the metrics above using the interventional distribution, $P(y|do(t))$, while regular machine learning approaches learn the conditional distribution $P(y|t)$.

It is also useful to mention that the method being evaluated specifically attempts to reduce confounding bias, but does not in any way solve the related issue of selection bias. While minimizing confounding ensures the estimated effects are correct for the sample population, known as internal validity, it cannot guarantee that these effects are generalizable outside of the sample, which is known as external validity Haneuse (2013). Techniques that address selection bias are outside of the scope, as they differ from those used to address confounding bias.

2.2 The Causal Effect Variational Autoencoder

The model being evaluated is the Causal Effect Variational Autoencoder (CEVAE). It attempts to infer unobserved confounders in a dataset from a set of proxy variables available in the dataset. Specifically, CEVAE is defined for data that follows the causal graph of Figure 1b, while compared to the standard graph used in causal effect estimation shown in Figure 1a.

CEVAE consists of two components: the inference network, and the generative network. These function as the encoder and decoder of the Variational Autoencoder (VAE), respectively. The goal of a VAE is similar to that of a regular autoencoder, which is a neural network that finds a lower dimensionality representation of data using fully connected layers that gradually decrease and then increase in dimension. The loss of such as model is the difference between the original and the reconstructed data, and the lower dimensional representation can be extracted from the middle layer which has fewest dimensions. A VAE



(a) Observed variables $\mathbf{x}$ confounding the treatment t and the outcome y.

(b) Unobserved variables $\mathbf{z}$ confounding t and y, with $\mathbf{x}$ being proxies for $\mathbf{z}$.

Figure 1: Causal graphs involving different variables and causal effects. Observed variables are shown with thicker borders than unobserved variables.

expands on this idea, with the encoder modelling the parameters of a distribution, and the decoder sampling from the distribution and reconstructing the input. For CEVAE, this distribution is the distribution of unobserved confounders. For more details on VAE's, refer to Kingma and Welling (2019).

CEVAE extends this to causal effect estimation using variational inference: neural networks model the distributions $q(t|\mathbf{x})$, $q(y|t, \mathbf{x})$, and $q(\mathbf{z}|t, y, \mathbf{x})$ from the proxies $\mathbf{x}$ in the encoder, to obtain an estimate of the latent distribution $p(\mathbf{z})$. Then, values are sampled and neural networks model the distributions $p(\mathbf{x}|\mathbf{z})$, $p(t|\mathbf{z})$, and $p(y|t, \mathbf{z})$. The authors show in Theorem 1 that these last three are sufficient to recover the interventional distribution, and therefore also the causal effect.

A notable part of the design by Louizos et al. (2017) is the model of the latent probability distribution $p(\mathbf{z})$, defined as the product of $D_{\mathbf{z}}$ normal distributions, where $D_{\mathbf{z}}$ is the dimensionality of the latent variable. If this dimensionality does not match that of the unobserved confounders in the dataset, the model is misspecified. For instance, in the case of a data generating process with a 3 dimensional normal latent variable, the modelled latent probability distribution should also be 3 dimensional. For conciseness, the number of dimensions in the latent variable of the data and the model are further referred to as data dimensionality and model dimensionality.

3 Methodology

In this section, the high level approach that was taken in order to answer the research question is described, along with a motivation. First, an overview of the approach is given in Section 3.1. Data is an integral part of evaluating causal inference models, so the choices made regarding this are discussed in Section 3.2. Section 3.3 motivates the use of evaluation metrics.

3.1 Overview

This research focuses on the performance of CEVAE when the model dimensionality does not necessarily correspond to the data dimensionality. In order to investigate this, three independent variables were identified: model dimensionality, data dimensionality, and data "complexity". It is not clear that the effects of these are independent of each other, so they are studied simultaneously, in a manner similar to a factorial design (Mukerjee and Wu, 2006). In this way, the effect of model latent dimensionality can be examined at different levels of latent confounder dimensionality and data "complexity". The three levels are explained below.

At the lowest level, the model dimensionality is varied, measuring the performance of CEVAE in some specific data configuration. This gives insight into which dimensionalities are best for that configuration and whether increasing the dimensionality improves the performance, or vice versa.

Then, the data dimensionality is alternated such that the configuration mentioned above is changed. In other words, the model dimensionality is varied for each data dimensionality. This gives insight into how the behaviours mentioned above change when the number of latent confounders in the data changes.

Finally, all of the above is considered one experiment, and is repeated for two different data "complexities". This term specifically refers to running CEVAE with data where the proxy variables are linear-Gaussian, versus with proxy variables that are nonlinear-Gaussian. This gives insight into whether the results obtained with one data generating process translate into similar conclusions with an alternate data generating process.

3.2 Data

Synthetic data is used for both of the two experiments due to the level of control over the data generating process it provides, and the availability of ground truths. Specifically, it is advantageous to be able to define the latent dimensionality and the distributions of variables, as these are two of the the independent variables. The use of semi-synthetic data such as the IHDP or Twins benchmarks was considered, but in this end this was not done because of decreased control over the proxy and latent distributions.

The data is defined in such a way that heterogeneous effects are created, that is, the effect depends on the value of the confounders. This increases the generalizability of the results, by removing the assumption of homogeneous treatment.

In all experiments, the treatment is binary. This is done for two reasons: First, as mentioned in Section 2.1, a binary treatment is a common case, as many investigations can be set up as treatment versus control. Second, it is a case not explored by the related research of Rissanen and Marttinen (2021), who critically evaluate CEVAE. Instead, their use of synthetic data focuses on entirely linear-Gaussian data, entirely binary data, and some semi-synthetic data.

It should be noted that the outcome variable is *not* linear-Gaussian for either of the two experiments mentioned in Section 3.1, as nonlinearities are required in order to create the heterogeneous effects mentioned in this section. The change mentioned there is specifically restricted to the proxies.

A drawback on synthetic data is that the overall generalizability of conclusions based on this type of data can sometimes be limited, as it can be argued that it is not representative of the datasets that would be encountered if the model was deployed. This is partially mitigated by performing the two experiments with two different data generating processes. Also, it helps that the second experiment introduces nonlinearities into the proxies, as these may exist in real-world datasets. If the conclusions for the two experiments are similar, this is a good indication that the conclusions do *not* only apply to a narrow scenario.

3.3 Evaluation Metrics

CEVAE is designed to infer the individualized treatment effect (ITE), defined by Equation (1). The most suitable metric to evaluate this is the square root of the precision in estimation of heterogenous effect (denoted further as $\sqrt{e_{\text{PEHE}}}$):

$$\sqrt{e_{\text{PEHE}}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\tau(\mathbf{x}_i) - \hat{\tau}(\mathbf{x}_i))^2}$$

Here, N is the dataset sample size, $\tau(\cdot)$ is the ground truth ITE, and $\hat{\tau}(\cdot)$ is the ITE estimated by the model. PEHE by itself, discussed by Cheng et al. (2022), is defined as the mean squared error in ITE. This penalizes large errors in ITE made by the model more than small errors, which is desirable, although in certain situations mean absolute error could be used. Taking the square root of PEHE ensures the error values have the same units and are therefore more intuitively comparable.

That being said, the error in average treatment effect (denoted further as e_{ATE}) is also considered:

$$e_{\text{ATE}} = |\mathbb{E}[\tau(\mathbf{x})] - \mathbb{E}[\hat{\tau}(\mathbf{x})]|$$

Often the two metrics chosen here agree on the performance of a model. Still, they are not redundant as considering both can help uncover certain undesirable results.

Finally, metrics are calculated on out-of-sample data according to the train/test split specified in Section 4.4, as generalizability of the model on unseen data from the sample population is important. Using within-sample data could unjustly improve the apparent performance of overfitting models. That being said, the fit of every model is still verified using the validation procedure mentioned in Section 4.4.

4 Experimental Setup

This section describes in detail the concrete experimental setup, including the data generation and configuration for each experiment in Sections 4.1 and 4.2, the model configuration in Section 4.3, and the evaluation loop in Section 4.4.

4.1 Linear-Gaussian Proxies with Binary Treatment

In order to control the latent confounder dimensionality in the data generating process as mentioned in Section 3.1, the following base data generating process with 3 latent variables is used, which follows the causal graph of Figure 1b. Below it is explained how this process was adapted for 1 latent variable. (Note that variables are identified with subscripts in front to prevent confusion with previously defined notation for instances):

$$ {}_1z \sim \mathcal{N}(1, 0.5) \qquad {}_2z \sim \mathcal{N}(-3, 0.5) \qquad {}_3z \sim \mathcal{N}(3, 1) \tag{3}$$

$$ {}_1x|_{1-3}z \sim \mathcal{N}({}_2z, {}_1w) \qquad {}_2x|_{1-3}z \sim \mathcal{N}({}_1z + {}_2z, {}_2w) \tag{4}$$

$$ {}_3x|_{1-3}z \sim \mathcal{N}({}_3z, {}_3w) \qquad {}_4x|_{1-3}z \sim \mathcal{N}({}_1z, {}_4w) \tag{5}$$

$$ {}_5x|_{1-3}z \sim \mathcal{N}({}_2z + {}_3z, {}_5w) \qquad {}_6x|_{1-3}z \sim \mathcal{N}({}_3z, {}_6w) \tag{6}$$

$$ {}_7x|_{1-3}z \sim \mathcal{N}({}_1z, {}_7w) \qquad {}_8x|_{1-3}z \sim \mathcal{N}({}_2z, {}_8w) \tag{7}$$

$$ {}_9x|_{1-3}z \sim \mathcal{N}({}_3z + {}_1z, {}_9w) \tag{8}$$

$$t|{}_{1-3}z \sim \mathrm{Bern}(\sigma({}_1z + {}_2z + {}_3z)) \tag{9}$$

$$y|t, {}_{1-3}z \sim \mathcal{N}(0.7{}_3z + 0.3{}_1z + \frac{1}{2}(t - 0.5)({}_1z + {}_2z), 0.1) \tag{10}$$

Where i is the data instance, $\sigma(\cdot)$ is the sigmoid function, ${}_{1-3}z$ are the latent distributions, ${}_{1-9}x$ are the proxies, ${}_{1-9}w$ are the proxy noise weights, t is the treatment, and y is the outcome. Non-zero latent means were used to ensure the causal effect is not zero. Values of the proxy noise weights are available in Appendix A.

Adapting this process for 1 latent variable is done in a way such that other factors likely to affect performance remained constant. Specifically, the number of proxies was kept the same, ensuring that CEVAE is fit on the same number of features. The following is changed: The 1 latent variable is distributed as ${}_1z \sim \mathcal{N}(0.8, 1)$, and all references to other latent variables in functions for the proxies, treatment, and outcome, are replaced with this new latent (e.g: the treatment for the 1-d case is $t|{}_1z \sim \mathrm{Bern}(\sigma({}_1z + {}_1z + {}_1z)))$.

Within each of the two cases in this experiment, the hypotheses are the following (in this paragraph, performance refers to both $\sqrt{e_{PEHE}}$ and e_{ATE}): Based on results obtained by the authors of CEVAE in a similar experiment, it is predicted that when the model dimensionality is higher than the data dimensionality, performance will degrade, as the model is overparameterized. When the model dimensionality is lower than the data dimensionality, performance should also degrade, because the model is not able to sufficiently model the latent space. It is not clear which case will lead to a worse degradation. The best performance should occur when the two dimensionalities are equal. The two metrics are not expected to differ in their trend.

Across the two cases, it is predicted that increasing the number of latent variables in the data will reduce performance for lower model dimensionalities, and improve performance for higher dimensionalities. Intuitively, the "window" within which the model performs well should shift higher. This follows from the idea that performance should improve when the dimensionalities are similar.

4.2 Nonlinear-Gaussian Proxies with Binary Treatment

Similarly to the data generating process above, data for the second experiment is specified as (Note that variables are identified with subscripts in front to prevent confusion with previously defined notation for instances):

$${}_1z \sim \mathcal{N}(1, 0.5) \qquad {}_2z \sim \mathcal{N}(-3, 0.5) \qquad {}_3z \sim \mathcal{N}(3, 1) \tag{11}$$

$${}_1x|{}_{1-3}z \sim \mathcal{N}({}_2z{}_3z - {}_1z, {}_1w) \qquad {}_2x|{}_{1-3}z \sim \mathcal{N}(\sigma({}_1z/{}_3z - 3), {}_2w) \tag{12}$$

$${}_3x|{}_{1-3}z \sim \mathcal{N}(2{}_3z + {}_1z + {}_1z, {}_3w) \qquad {}_4x|{}_{1-3}z \sim \mathcal{N}(\sigma({}_1z{}_2z), {}_4w) \tag{13}$$

$${}_5x|{}_{1-3}z \sim \mathcal{N}({}_2z{}_1z + 2, {}_5w) \qquad {}_6x|{}_{1-3}z \sim \mathcal{N}(0.2{}_3z + 2{}_2z + {}_1z, {}_6w) \tag{14}$$

$${}_7x|{}_{1-3}z \sim \mathcal{N}(\sin({}_1z{}_1z), {}_7w) \qquad {}_8x|{}_{1-3}z \sim \mathcal{N}({}_2z/{}_3z - 5, {}_8w) \tag{15}$$

$${}_9x|{}_{1-3}z \sim \mathcal{N}({}_3z + {}_1z + {}_2z, {}_9w) \tag{16}$$

$$t|{}_{1-3}z \sim \mathrm{Bern}(\sigma({}_1z + {}_2z + {}_3z)) \tag{17}$$

$$y|t, {}_{1-3}z \sim \mathcal{N}(0.7{}_3z + 0.3{}_1z + \frac{1}{2}(t - 0.5)({}_1z + {}_2z), 0.1) \tag{18}$$

Where the notation, proxy weights, and the procedure to construct the 1 dimensional latent version, are the same as in Section 4.1. The only change is the addition of nonlinearities in the proxy functions.

For the two cases in this experiment, the hypotheses are the same as for the previous experiment, as nonlinearities are not expected to change the interaction between the dimensionalities. The key contribution of this second experiment is verifying that these interactions do not change across two different datasets.

4.3 Model

An implementation of CEVAE from the Pyro probabilistic modelling framework[1] is used, implemented in Python and developed by Bingham et al. (2018). This implementation exposes an interface to set the hyperparameters of the model, which allows for varying the model dimensionality.

Hyper-parameters for the model are chosen initially based on Louizos et al. (2017), and then changed slightly through experimentation on data unrelated to the experimental data. Running the full evaluation on multiple hyperparameters was infeasible. The goal was to balance the need for sufficient learning capacity with the risk of overfitting. Unless otherwise stated, the model is configured with: a 1-dimensional normal latent space, 3 hidden layers with 200 dimensions, 100 samples of the latent distribution, a batch size of 1000, 100 epochs of training, and a normal outcome distribution. The optimizer used was the default `ClippedAdam`[2] from the Pyro framework, with a learning rate of 1e-2, a learning rate decay of 0.01, and a weight decay of 1e-4.

4.4 Evaluation Loop

The evaluation loop is important in ensuring that the model is fit correctly, and that the subsequent inference is done on correct data.

A train/test split of 0.8/0.2 is used, and the model is validated on the train and test sets during training in order to track the model loss and prevent overfitting. Normally this would require a separate validation set, but in this case early stopping was not used. Model loss was tracked not for optimization, but to ensure the model training has occurred correctly.

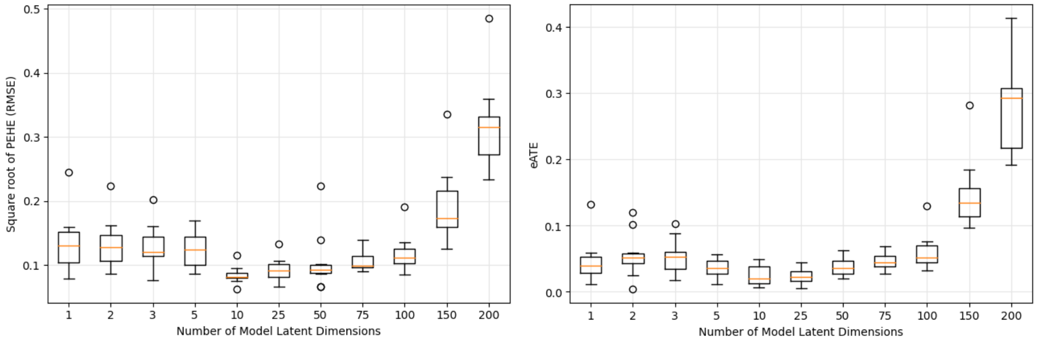
When running the model against some data configuration, 10 replications are run in order to increase reliability of the results, and provide statistics such as the mean and quartiles. Each of these replications is seeded with 10 predetermined values, available in Appendix A, which makes the metrics easier to compare between model dimensionalities (For example, each model run shown on the x-axis of Figure 2a uses 10 different datasets that do not change when model dimensionality changes.). That being said, the optimization and sampling within the VAE is still a source of randomness.

5 Results

This section describes the findings of the aforementioned experiments. Section 5.1 contains results of the linear-Gaussian experiment, and Section 5.2 contains results of the nonlinear-Gaussian experiment. Section 5.3 contains small scale verification experiments used to validate any hypotheses that arose from preceding results. Each of these sections contains

[1] Available at `https://github.com/pyro-ppl/pyro`

[2] Source code available at `https://docs.pyro.ai/en/stable/_modules/pyro/optim/clipped_adam.html`



(a) $\sqrt{e_{\text{PEHE}}}$ against model latent dimensionality.

(b) e_{ATE} against model latent dimensionality.

Figure 2: Dimensionality experiment for 1 latent variable and 9 linear-Gaussian proxies.

a small reflection on the results obtained as compared to the hypotheses established in Sections 4.1 and 4.2. Many of these results are presented in the form of boxplots, which show the five-number summary in a graphical form. Unless specifically mentioned, outliers are not taken into consideration in the explanations that follow.

5.1 Misspecification for Linear-Gaussian Proxy Synthetic Data

5.1.1 1 Latent Normal Dimension

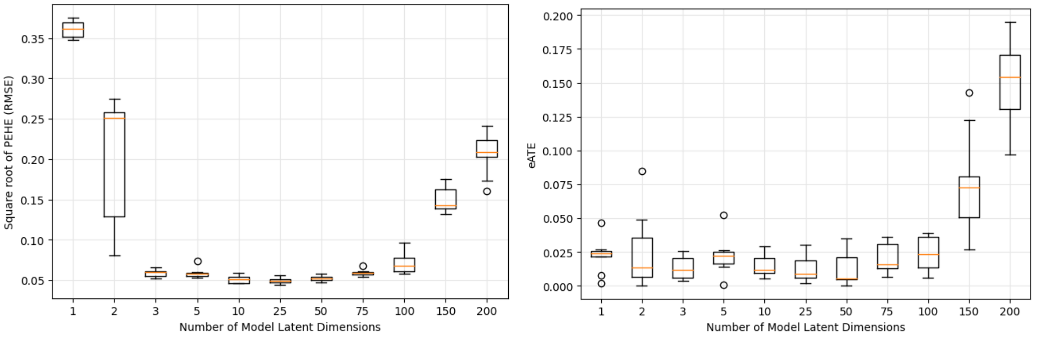
Figures 2a and 2b show results for varying the model latent dimensionality for the synthetic experiment with a 1 dimensional latent normal distribution in data with linear-Gaussian proxies.

It is evident that model performance in terms of both metrics is best for model dimensionalities of 10, 25, and 50. This is both in terms of a low mean, and a small interquartile range. That being said, all values between 1 and 100 demonstrate relatively similar performance, with errors only increasing significantly for values of 150 and 200. It was expected that performance would decrease as the model dimensionality becomes larger than the data dimensionality, but this happens at a surprisingly high value. The two metrics do not deviate in their trends, which was expected, but the best performance does not occur when the dimensionalities are equal. This was expected to occur at 1, but results show the best performance is at 10.

5.1.2 3 Latent Normal Dimension

Figures 3a and 3b show results for varying the model latent dimensionality for the synthetic experiment with a 3 dimensional latent normal distribution in data with linear-Gaussian proxies.

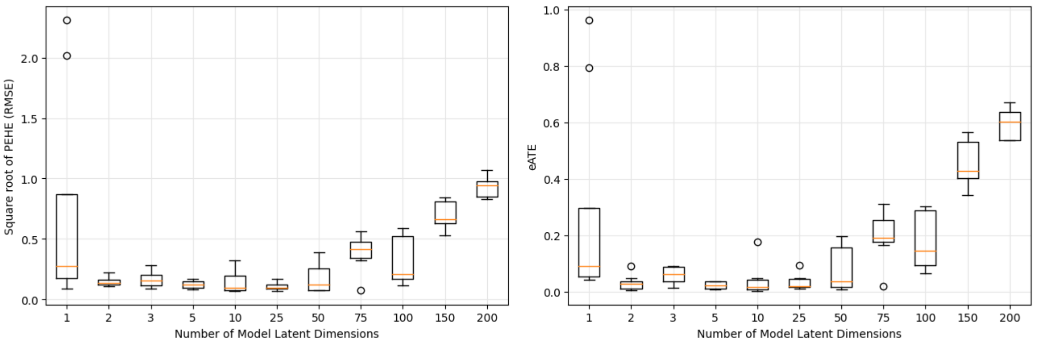
Increasing the data dimensionality resulted in noticeable differences. At low model dimensionality such as 1 and 2, it is observed that $\sqrt{r_{PEHE}}$ increases, which was expected. e_{ATE} did not increase similarly, which was not expected. The decrease in performance when the model dimensionality is lower than the data dimensionality is also expected, although it was not predicted that this increase in error would be this severe for the lower dimension. The hypothesis of whether the "window" within which the model performs well will shift up



(a) $\sqrt{e_{\text{PEHE}}}$ against model latent dimensionality.

(b) e_{ATE} against model latent dimensionality.

Figure 3: Dimensionality experiment for 3 latent variables and 9 linear-Gaussian proxies.



(a) $\sqrt{e_{\text{PEHE}}}$ against model latent dimensionality.

(b) e_{ATE} against model latent dimensionality.

Figure 4: Dimensionality experiment for 1 latent variable and 9 nonlinear-Gaussian proxies.

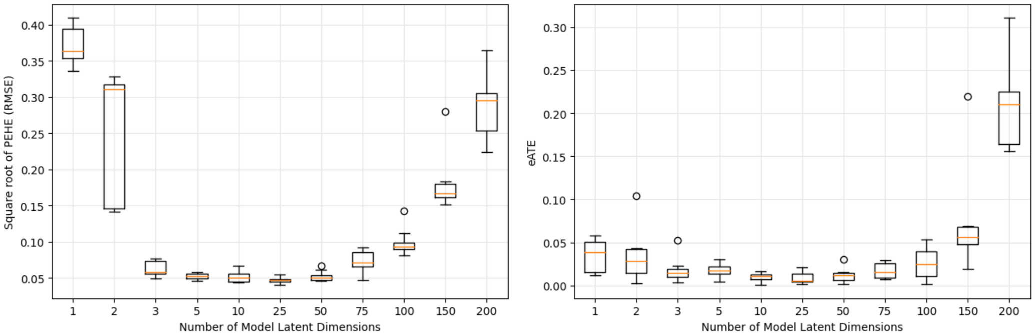
cannot be clearly read from the graph, although the interquartile range of the runs between 3 and 75 dimensions quite clearly decreases in width. The disagreement between metrics is investigated further in Section 5.3.1.

5.2 Misspecification for Nonlinear-Gaussian Proxy Synthetic Data

5.2.1 1 Latent Normal Dimension

Figures 4a and 4b show results for varying the model latent dimensionality for the synthetic experiment with a 1 dimensional latent normal distribution in data with nonlinear-Gaussian proxies. These results are compared to their linear-Gaussian counterpart in Section 5.1.1.

Overall, a similar trend is seen of increasing error as model dimensionality changes further from the data dimensionality, but there are two differences. First, this increase in error is already noticeable around 75 dimensions, compared to the 150 dimensions in the previous experiment. Second, an increased variance in the error of model dimensionality 1 seems to be counter intuitive, as this is dimensionality which was predicted to have the best performance.



(a) $\sqrt{e_{\text{PEHE}}}$ against model latent dimensionality.

(b) e_{ATE} against model latent dimensionality.

Figure 5: Dimensionality experiment for 3 latent variable and 9 nonlinear-Gaussian proxies.

5.2.2 3 Latent Normal Dimension

Figures 5a and 5b show results for varying the model latent dimensionality for the synthetic experiment with a 3 dimensional latent normal distribution in data with nonlinear-Gaussian proxies. These results are compared to their linear-Gaussian counterpart in Section 5.1.2.

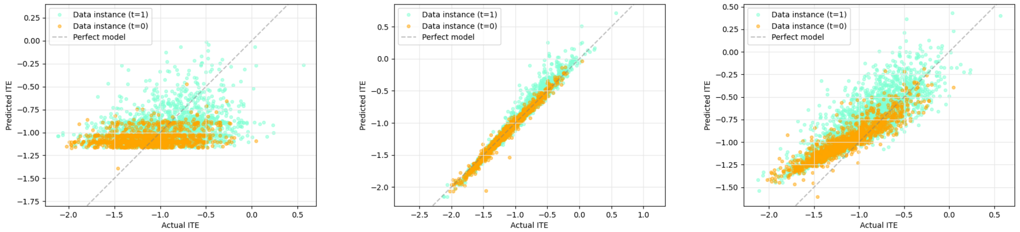
The differences between these results and those using linear-Gaussian proxies are few. Certain means and quartiles do differ, but overall we see a similar trend. Considering these results and those of Section 5.2.2, there is an indication that the overall behaviour of varying dimensionality translates to nonlinear-Gaussian proxy data.

5.3 Additional Verification Experiments

5.3.1 Metric Disagreement

As seen in Sections 5.1.2 and 5.2.2, the $\sqrt{e_{PEHE}}$ and e_{ATE} metrics do not follow similar trends, which is surprising and warrants a small investigation. This also provides a chance to examine the high-dimensionality case more. Plots of estimated ITE against ground truth ITE, shown in Figure 6, were made for the 3 models that showed interesting behaviour in Figure 5a. This is the case of 1, 3, and 200 model dimensions.

Figures 6a to 6c show respectively the inference results for these models. These plots given insight into why the 1-d and 200-d models are underperforming. Figure 6a seems to simply lack the capacity to model the complexity of the latent space, instead able to only optimize the ATE, which is the average. Figure 6c Lags behind in both metrics, so there are two possibilities: the model is overfitting, or there is not enough data considering the large number of latent dimensions to tune. One more mini-experiment is repeated to investigate this in Section 5.3.2 It is also interesting that the predictions occur in two distinct groupings, which can be explained by the fact that CEVAE uses a distinct neural network for each treatment value, in order to better handle data with imbalanced treatments. Formally, $P(y|do(t=1))$ and $P(y|do(t=0))$ are modelled separately.



(a) Using a 1 dimensional latent, the model incorrectly estimates the ITE for most instances, yet the eATE is small.

(b) Using a 3 dimensional latent, data instances lie close to the perfect model line.

(c) Using a 200 dimensional latent, the data shows hints of fitting to the perfect fit line, but is quite spread out.

Figure 6: Plots of estimated ITE against ground truth ITE For different model latent dimensions in the experiment with nonlinear-Gaussian proxies and 3 latent confounders in the data. The true ATE for this plot is close to -1. Instances for which $t = 1$ are shown in green, those for which $t = 0$ are shown in orange, and a perfect model line indicating a ideal estimation of ITE is shown in grey.

5.3.2 Increasing Sample Size for Extreme Dimensionality Cases

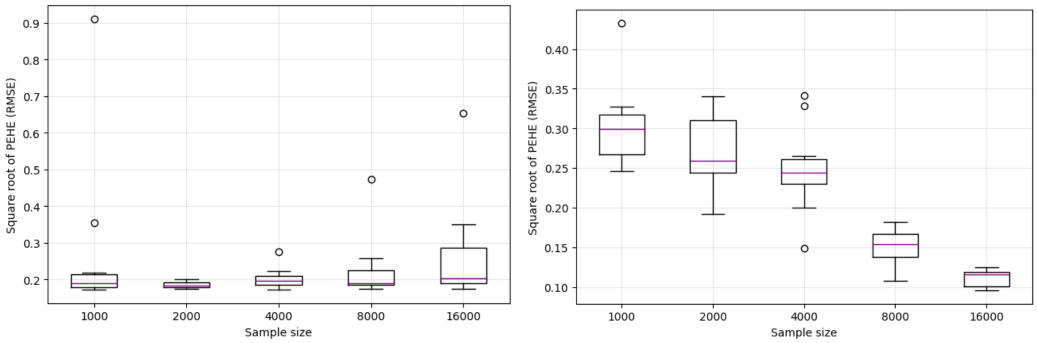
To verify why the 1-d and 200-d cases are underperforming, these models are trained on gradually increasing samples sizes in order to find out if their errors decrease. Every plot shows signs of high dimensionality models underperforming, and this can uncover the reason why. Figures 7a and 7b show the results of this verification.

As expected, the 1-d model cannot improve its performance on increasing sample sizes, supporting the idea that it is not powerful enough to model the 3 latent confounder. The 200-d model, on the other hand, clearly improves as the sample size increases, even resulting in a lower error than the 1-d model. This indicates overfitting is the cause of the error identified in previous experiments. Although this experiment was done with nonlinear-Gaussian proxies, its conclusion should translate to linear-Gaussian proxies, given that the experiment in Section 5.2.2

6 Discussion

Several hypotheses were established in Sections 4.1 and 4.2, and some of these are supported by the results obtained in Section 5. Sections 5.1.1 and 5.1.2 show that the performance does not degrade as quickly as expected when the model dimensionality is higher than the data dimensionality. Still, eventually performance does degrade when the dimensionalities differ by a lot. When the data generating process has more latent variables than CEVAE is able to model, performance also decreased, very rapidly.

The use of two error metrics allowed for the identification of an interesting case in Section 5.1.2, where $\sqrt{e_{PEHE}}$ was high, yet e_{ATE} remained low. This was investigated further in Section 5.3.1, making it seem like the issue was an underfitting model. By following up with a verification experiment that increased sample size for the problematic models, Section 5.3.2 showed clearly that models with too few dimensions suffer from underfitting and cannot improve with more data, while models with too many dimensions are overfitted,



(a) Increasing sample size with 1 model dimensionality. The configuration corresponds to the one used for Figure 6a.

(b) Increasing sample size with 200 model dimensionality. The configuration corresponds to the one used for Figure 6a.

Figure 7: Error against increasing sample size for the two extreme cases of underperforming models

which could be solved in this case using more data.

The repetition of the experiments with two distinct data generating processes, the linear-Gaussian and the nonlinear-Gaussian data, indicated that the overall behaviour of CEVAE under misspecification does not differ between the two datasets. This indicates some level of generalizability, which should be investigated further using semi-synthetic data.

On the other hand, some hypotheses were rejected by the results. In Sections 5.1.2 and 5.2.2, it could not be confirmed that increasing the latent dimensionality of the data *simultaneously* reduces performance for lower model dimensionality, and improves performance for higher dimensionalities. It is likely that this is because the increase was from 1 to 3, which is simply not sufficient to improve performance of a model with 150 dimensions.

The idea that the best model performance will occur when the model dimensionality and the latent dimensionality are equal also could not be confirmed, as often the best results were obtained when the model had up to 10 more dimensions than the data. This could be a statistical issue caused by the relatively small number of replications, but a likely explanation is that CEVAE simply does not always perform best when the two dimensionalities are equal. Upon further investigation, evidence for this was found by Rissanen and Marttinen (2021), who show using a synthetic dataset that, under certain conditions, the model requires more dimensions than the number data dimensions.

It is also worthwhile to discuss other limitations of this research. Although the results obtained applied to both datasets, it is possible that these conclusions do not apply to certain dataset. A key limitation in performing this evaluation was the computational power available to train CEVAE, due to which only 10 replications could be run for each configuration. Lastly, the implementation of CEVAE used was checked to ensure it adheres to the specification of the original CEVAE paper, but it is also possible that slight differences in the implementation had an impact on the results obtained.

7 Conclusions

The central question of this research was to investigate the performance of the CEVAE model when its latent variable is not defined with the same number of dimensions as the data latent variable. The scope of the investigation is narrowed down by focusing on two datasets. The first case, called the "nonlinear-Gaussian proxy dataset", consists of a binary treatment, a nonlinear-Gaussian outcome, linear-Gaussian latent variables, and nonlinear-Gaussian proxies. The second case, called the "linear-Gaussian proxy dataset", consists of a binary treatment, a nonlinear-Gaussian outcome, linear-Gaussian latent variables, and linear-Gaussian proxies.

It is found that the CEVAE model is surprisingly resistant to performance degradation when the model's latent variable is higher than the latent variable of the data. Increasing the sample size resolves this issue, even when the model dimension is much higher (200) than the the latent dimension (3). On the other hand, performance suffers when the model dimension is lower than the latent dimensions. This cannot be solved by fitting to more data, as CEVAE is unable to correctly model the latent space. It is also found that CEVAE does not necessarily perform best when the two dimensionalities correspond, due to the fact that the latent space serves more as an estimator rather than a true model. Overall, results indicate that it is a safer option to specify CEVAE with more dimensions, when in doubt.

8 Future Work

Several potential improvements and extensions have been identified as the research was carried out. Applying CEVAE to more datasets, and using semi-synthetic data, would be beneficial. Semi-synthetic data is more generalizable due to the more realistic distributions of covariates. The experiment can also be repeated with more replications, in order to obtain more reliable mean and variance information. Lastly, it would be useful to apply this research to the latent distribution of the CEVAE. This was not done to manage the scope, but is an equally important component of misspecification for the model.

9 Responsible Research

This section reflects on the ethical aspects of this research, as well as the reproducibility of the outlined methodology respectively in Sections 9.1 and 9.2.

9.1 Ethical Implications

Examples of experiments used in causal effect estimation literature are often from the medical domain, and for good reason. Some of the most direct and obvious applications of this technique are in the estimation of the effect of drugs and other interventions on the health status of patients. This information can be used to reject or approve various medical interventions that can potentially be anywhere from life-saving to fatal. Machine learning techniques such as CEVAE could also potentially be used in critical infrastructure.

For all of these reasons, it is vital that the results, discussions, and conclusions in this research document were communicated honestly and correctly, including all relevant assumptions, limitations, and disadvantages of the methods presented. Beyond practical applications, the academic community relies on the intellectual honesty of its members to prove it as a reliable source of knowledge in the eyes of the public.

9.2 Reproducibility

Reproducibility is fundamental for good quality science, and this is done by describing in detail the steps taken and the tools used. Sections 3 and 4 respectively describe the steps taken for this research paper from both a conceptual and concrete perspective, allowing for the results obtained to be reproduced. This also includes relevant hyperparameters, evaluation procedure, software packages, data generation, data seeding, and the source code for the entire evaluation. The repository[3] containing the code also is made public under the MIT license. This a permissive, open source license that allows anyone to benefit from, expand on, and verify/challenge this research.

That being said, risks associated with reproducibility can be minimized, but not eliminated. Only 10 replications were used for experiments shown in Section 5, due to limitations on computing power combined with the large number of model dimensions tested. There are also remaining sources of randomness within the training process, such as model initialization, optimization, and latent dimension sampling, which cannot be eliminated. That being said, these limitations do not make the results irreproducible.

References

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P. A., Horsfall, P., and Goodman, N. D. (2018). Pyro: Deep universal probabilistic programming. *CoRR*, abs/1810.09538.

Cheng, L., Guo, R., Moraffah, R., Sheth, P., Candan, K. S., and Liu, H. (2022). Evaluation methods and measures for causal learning algorithms.

Guo, R., Cheng, L., Li, J., Hahn, P. R., and Liu, H. (2018). A survey of learning causality with data: Problems and methods. *CoRR*, abs/1809.09337.

Haneuse, S. (2013). Distinguishing selection bias and confounding bias in comparative effectiveness research. *Medical care*, Publish Ahead of Print.

Johansson, F. D., Shalit, U., and Sontag, D. (2016). Learning representations for counterfactual inference.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *CoRR*, abs/1906.02691.

Kuroki, M. and Pearl, J. (2014). Measurement bias and effect restoration in causal inference. *Biometrika*, 2.

Louizos, C., Shalit, U., Mooij, J., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models.

Mukerjee, R. and Wu, C.-F. (2006). *A modern theory of factorial design.* Springer.

Pearl, J. (2009). *Causality.* Cambridge University Press, 2 edition.

[3] Available at `https://github.com/patrikbartak/cevae-evaluation`

Pearl, J. (2012). On measurement bias in causal inference.

Pham, T. T. and Shen, Y. (2017). A deep causal inference approach to measuring the effects of forming group loans in online non-profit microfinance platform.

Rissanen, S. and Marttinen, P. (2021). A critical look at the identifiability of causal effects with deep latent variable models. *CoRR*, abs/2102.06648.

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

Shalit, U., Johansson, F. D., and Sontag, D. (2016). Estimating individual treatment effect: generalization bounds and algorithms.

A Data generating parameters

Table 1: Proxy noise weights used for linear-Gaussian and nonlinear-Gaussian synthetic data.

Weight	Value
${}_1w$	0.04
${}_2w$	0.17
${}_3w$	0.19
${}_4w$	0.10
${}_5w$	0.11
${}_6w$	0.13
${}_7w$	0.22
${}_8w$	0.14
${}_9w$	0.08

Table 2: Values used to seed the Pyro distribution functions.

Replication	seed
1	77295
2	40865
3	82635
4	86576
5	84055
6	15487
7	60076
8	39714
9	60063
10	46530