Non-factorised identifiable variational autoencoders for causal discovery and out-of-distribution generalisation

Andrei Tiberiu Alexandru

Abstract

In many situations, given a set of observations, we would like to find the factors which cause or influence the data we observe. To learn these factors, we can use a deep generative model such as a variational auto-encoder (VAE), which maps the data to a latent space. However, a limitation of the VAE is that it is not identifiable, in the sense that two different sets of parameters may yield the same model. To address this, it is possible to augment the VAE in such a way that it becomes identifiable while remaining capable of flexible representations. This paper explores invariant causal representation learning (ICaRL), an algorithm for causal representation learning with possible applications in causal discovery and out-of-distribution generalisation.

1 Introduction

Causal inference [Pea09] is a paradigm at the intersection of statistics, social science and machine learning concerned with the study of cause and effect. It is often framed as a more nuanced alternative to statistical inference, the distinction being that the latter is only capable of capturing correlation between random variables, whereas the former can make statements about causes. One of the most important applications of causal inference is counterfactual reasoning, a way of asking the question: "what would happen if?" In this sort of analysis, data that is collected under one regime is used to estimate a quantity of interest under a slightly different regime. The difference between the two regimes is usually given by an intervention – the act of modifying a variable to some known value to observe the downstream effects.

One of the most widely used formalisms in causal inference is that of a structural causal model (SCM) and its representation as a causal graph. A structural causal model is a set of equations of the form

$$\mathcal{S}_i : X_i \leftarrow f_i(Pa(X_i), N_i)$$

where X_i are random variables and $Pa(X_i)$ are its parents – the random variables that cause X_i — and N_i are independent noise random variables [Arj20]. An intervention is then modelled as replacing one or several of the equations

$$\mathcal{S}_i^e : X_i^e \leftarrow f_i^e(Pa(X_i^e), N_i^e)$$

[PBM16] considers there to be two broad categories of interventions: do-interventions which set the value of a variable to a fixed artifical value (corresponding to the do-calculus [Pea09] interpretation of an intervention) and soft interventions that are alterations to the noise component.

SCMs are equivalent to causal graphical models, represented as directed acyclic graphs $G = (V, E)$ comprising a set of nodes or vertices and edges which connect them. Figure 1 shows an example of a causal graph and its associated SCM, both depicting a causal structure we'll return to in Section 2. Interventions on causal graphs are sometimes called "mutilations", because they equate to various changes to the graph structure, such as deleting or adding edges.

While interventions are a useful concept for thinking about changes to the causal structure, in many cases it's impossible or unethical to perform an intervention. Furthermore, we sometimes don't even know the SCM or underlying causal graph, and yet we would like to estimate some quantity counterfactually. In these situations, it would be useful to infer an underlying causal structure from data that has already been observed, a process known as causal discovery.

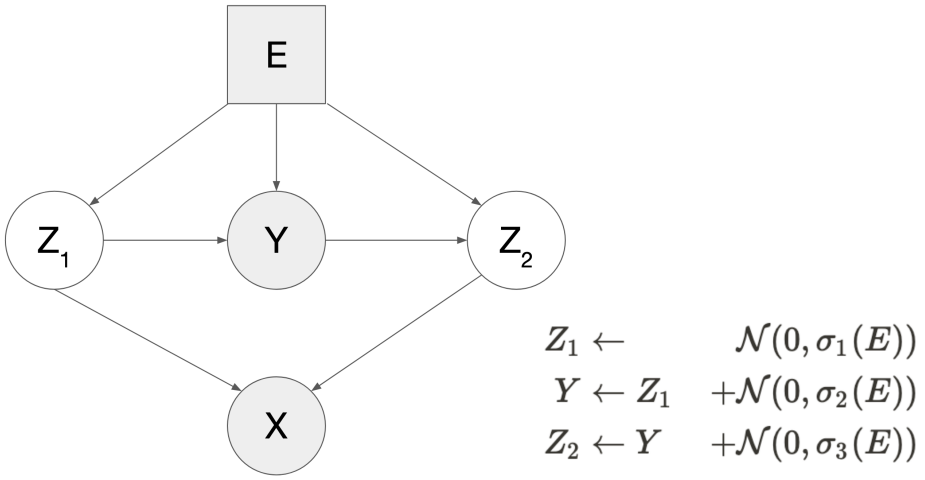

Figure 1: A causal graph and its associated structural causal model. In this example, E is a deterministic environment variable, X is an observed random variable that arises from a combination of Z_1 and Z_2, and Y is another observed variable. The functions σ_i introduce noise into the structural equation model, and their output depends on the environment.

Identifying the variables in a causal graph is conceptually similar to independent component analysis (ICA)[Com94]. Given an observed variable Y and a set of unobserved variables $\mathbf{X}$ which cause Y in some form, ICA tries to find the inverse of the mapping

$$f(\mathbf{X}) = Y$$

For linear ICA, we can express f as a linear combination of the sources X:

$$f(\mathbf{X}) = \mathbf{M}\mathbf{X} + \mathbf{v}$$

where $\mathbf{v}$ is noise. For nonlinear ICA, f is an unknown invertible transformation of the sources. This is very closely related to deep generative models, so a common approach is to train a deep generative model like a variational autoencoder (VAE) [KW13] to reconstruct the data in an unsupervised fashion. The VAE maps each input to a latent space, then generates an output based on a sample of latent variables from the space. This topic is sometimes also called disentanglement [SvKT+21].

A limitation of the VAE is that the model is not identifiable. Identifiability of a model is defined by [KKMH20] as

$$\forall(\boldsymbol{\theta}, \boldsymbol{\theta}') : p_{\boldsymbol{\theta}}(\mathbf{x}) = p_{\boldsymbol{\theta}'}(\mathbf{x}) \implies \boldsymbol{\theta} = \boldsymbol{\theta}'$$

If the VAE were identifiable, after training the model to approximate $p_{\theta^*}(x|z)$, we could reuse θ^* to calculate the true prior over latents $p_{\theta^*}(z)$ and the conditional $p_{\theta^*}(z|x)$, which is what we really care about from a disentanglement perspective: we want to know the distribution over latents z that influence our data.

One improvement to the VAE comes in the form of the identifiable variational autoencoder (iVAE) [KKMH20], which uses a factorised prior over latents z. The model assumes another observed variable $\mathbf{u}$, which could be "for example, the time index in a time series, previous data points in a time series, some kind of (possibly noisy) class label, or another concurrently observed variable" [KKMH20] It then uses as prior $p(\mathbf{z}|\mathbf{u})$ instead of $p(\mathbf{z})$.

Out-of-distribution (OOD) generalisation refers to the ability of models to generalise when the data shifts in some systematic way across multiple environments of interest $\mathcal{E}$. OOD is a challenge for neural networks because the training data is assumed to be independent and identically distributed – an assumption which does not always hold. In practice, neural networks have been observed to learn spurious correlations that leads to failing in surprising ways on slightly different data. A canonical example comes from [BVHP18], where an image recognition network was found to misclassify cows on

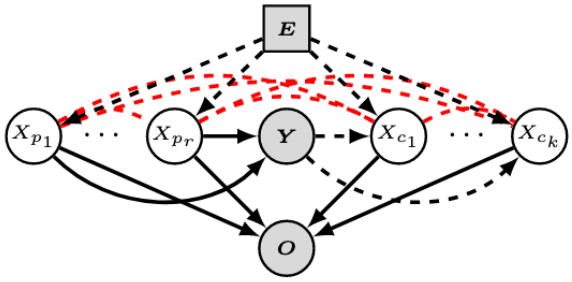

Figure 2: The causal graph blueprint used by ICaRL. Observed variables are shaeded grey: E is an environment variable, Y is a target variable of interest, and O is the observed data that is assumed to arise as a combination of latent variables X_{p_i}, X_{c_j}. Solid edges are relationships likely to be invariant across environments, whereas the dotted relationships may or may not hold from a domain to the other. Figure from [LWHLS21].

beaches as camels, because it was learning the correlation cow - grass, camel - sand. We want to be able to detect when a network picks up on this sort of correlation, or ideally prevent it altogether.

The main insight that brings together OOD generalisation, causal inference and independent component analysis is that fundamental characteristics and mechanisms that we want networks to learn are invariant across environments. In the earlier example, a cow is invariant to the background of the image – it should still be classified as a cow. From a causality perspective, invariant features tend to be causal features – factors which give rise to the data. Learning these features means that a network can combine them in arbitrary ways to recognise new data [SvKT+21].

One notable approach for learning invariant representations is the invariant risk minimisation (IRM) algorithm [ABGLP19]. It simultaneously learns a representation of the data, $\phi(x)$ and a classifier w^* such that $w \circ \phi$ is optimal across all environments. IRM has its limitations – mainly that in this form, the optimisation task is prohibitively difficult – so more and more research is focussing on simple ways to find the invariant representation ϕ.

In "Invariant Causal Representation Learning for Out of Distribution Generalization" [LWHLS21], the authors remove the conditional factorised prior requirement and recover an identifiable VAE with a more general conditional prior belonging to an exponential family. This allows them to recover the causes of a target variable Y in such a way that they can train an invariant predictor similar to IRM. In this paper, I partially replicate the results of the non-factorised identifiable VAE from a causal discovery perspective on synthetic data.

2 Preliminaries

The variational auto-encoder (VAE) [KW13] is a deep latent variable model of the form:

$$p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})p_{\boldsymbol{\theta}}(\mathbf{z})$$

where $\boldsymbol{\theta}$ is a vector of parameters, $\mathbf{z}$ is a vector of latent variables and $\mathbf{x}$ are the observed data. The VAE jointly trains a generative model of this form alongside an inference model $q_\phi(\mathbf{z}|\mathbf{x})$ that approximates the posterior $p_{\boldsymbol{\theta}}(z|x)$ by maximising the evidence lower bound (ELBO) of the dataset $\mathcal{D}$:

$$\mathcal{L}(\boldsymbol{\theta}, \phi) := E_{q_\mathcal{D}}\Big[E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})]\Big]$$

The identifiable VAE [KKMH20] additionally assumes that the prior $p(\mathbf{z})$ is conditionally factorised over an additional observed variable $\mathbf{u}$, i.e. that its probability density function is given by:

$$p_{\mathbf{T},\boldsymbol{\lambda}}(\mathbf{z}|\mathbf{u}) = \prod_i \frac{Q_i(z_i)}{Z_i(\mathbf{u})} \exp\left[\sum_{j=1}^{k} T_{ij}(z_i)\lambda_{ij}(\mathbf{u})\right]$$

where Q_i is the base measure, $Z_i(\mathbf{u})$ is the normalising constant, $\mathbf{T}_i$ are the sufficient statistics and $\boldsymbol{\lambda}_i(\mathbf{u})$ the corresponding parameters which depend on the additional observed variable $\mathbf{u}$.

3 Invariant causal representation learning (ICaRL)

Invariant causal representation learning (ICaRL) [LWHLS21] is a three-part algorithm. It operates on a general causal graph (Figure 2) and, under certain assumptions, provides guarantees regarding the generalisation capability and identifiability of the algorithm. The three phases are as follows:

1. Recovering latents with a **non-factorised identifiable VAE (nf-iVAE)**. This is a modification to the identifiable VAE such that the conditional prior over the latents is no longer required to be factorised. This is the main contribution of the ICaRL paper, in which they show that relaxing the condition still yields an identifiable VAE under certain assumptions. Specifically, the nf-iVAE assumes a prior from a general exponential family, the idea being that a more flexible distribution can capture arbitrary dependencies between the latents. An exponential family is a set of distributions whose probability density functions can be expressed as:

$$p(\mathbf{X}) = \frac{\mathcal{Q}(\mathbf{X})}{\mathcal{Z}(\boldsymbol{\theta})} \exp(\langle \mathbf{T}(\mathbf{X}), \boldsymbol{\theta} \rangle)$$

   The nf-iVAE is used to estimate a distribution over latent variables given the data, which is fed into the next phase.

2. Discovering direct causes. In the second phase, a skeleton graph is recovered from the latents using the Peter-Clark algorithm [SGSH00], which is used to identify the neighbours of a variable of interest Y. Given $\mathbf{Ne}(\mathbf{Y})$, we want to discover the parents of Y, $\mathbf{Pa}(\mathbf{Y})$. This is done through the kernel conditional independence test [ZPJS12]. The idea behind the test is that given two variables Z_1, Z_2, their conditional dependence increases when we additionally condition on Y, *only when they both cause* Y. The special case where the target variable has only one cause is addressed separately.

3. Training an invariant predictor. Finally, given an invariant representation of Y in the form of its parents $\mathbf{Pa}(\mathbf{Y})$, a predictor is learned which is expected to be optimal across environments according to the IRM principle [ABGLP19]. This means that the predictor should generalise well out-of-distribution.

4 Experiments

I ran two sets of experiments, one on synthetic data and the other on the dSprites dataset [MHHL17]. The hyperparameters are the same on both experiments. Specifically, no hyperparameter optimisation was done – the values used are the defaults from the implementation of iVAE [KKMH20].

4.1 Synthetic data

The synthetic data experiment follows [LWHLS21] and uses the following structural causal model:

$$\begin{aligned} Z_1 &\leftarrow \mathcal{N}(0, \sigma_1(E)) \\ Y &\leftarrow \mathcal{N}(0, \sigma_2(E)) + Z_1 \\ Z_2 &\leftarrow \mathcal{N}(0, \sigma_3(E)) + Y \end{aligned}$$

The data is generated from a simplified model, following Appendix K in the ICARL paper [LWHLS21], i.e. that $\sigma_1 = 1, \sigma_2 = 0, \sigma_3 = \{0.2, 2, 100\}$, with the final value of σ_3 used for testing. We draw 1000 samples for Z_1, Y, Z_2 for each of the three environments. This simple construction is easy to study because we can use the environment variable E as the additional observed variable required by the iVAE.

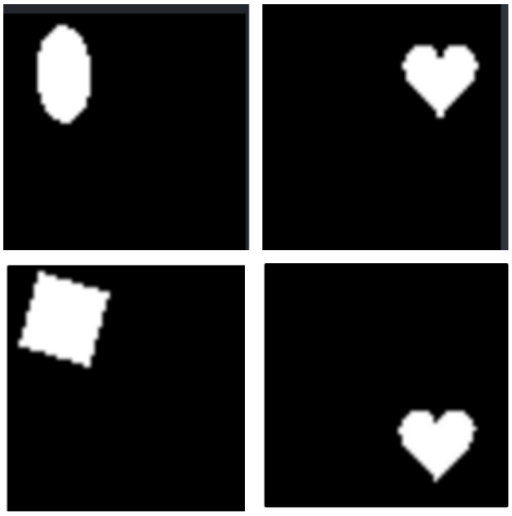

Figure 3: An example of the images in the dSprites [MHHL17] dataset. There are three shapes whose size, orientation and position in the frame change.

4.2 dSprites

The motivation for using dSprites comes from the paper "Visual representation learning does not generalize strongly in the same domain" [SvKT+21]. There, authors find that many common neural network architectures are largely unable to recover the latents (or sources, from an ICA perspective) that give rise to the observed data. One of the models they benchmark, the β-VAE [HMP+16], is very similar to the approach used by the iVAE , so it would be interesting to see whether the identifiability aspect of the latter enables it to do generalise better.

The dSprites dataset is a set of 64x64 images containing a white shape on a black background Figure 3. Each image is generated through a combination of 5 latent factors:

- Shape: square, ellipse, heart
- Scale: 6 values linearly spaced in [0.5, 1]
- Orientation: 40 values in [0, 2 pi]
- Position X: 32 values in [0, 1]
- Position Y: 32 values in [0, 1]

With all possible combinations contained in the dataset.

In [SvKT+21], the data is split systematically along these factors of variation to test interpolation, extrapolation and composition. In my experiments, the training set contains all images except the ones where the shape is in the bottom-right corner, and the test set contains just the latter. What I would like to see here is that the trained nf-iVAE has the capacity to infer the correct latents given an image from the test set. Specifically, it should map a test image to a set of latent values such that when an image is generated with those latent values, it matches the initial test image.

For dSprites, there is no additional observed variable $\mathbf{u}$, so one must be generated somehow such that we can apply the nf-iVAE with a conditional prior. I decided to use one of the true latents, the shape (circle, heart, square). The downside of this is that it reveals information about the dataset, effectively making the problem easier; the upside is that we can now use the nf-iVAE to try and disentangle the other latents.

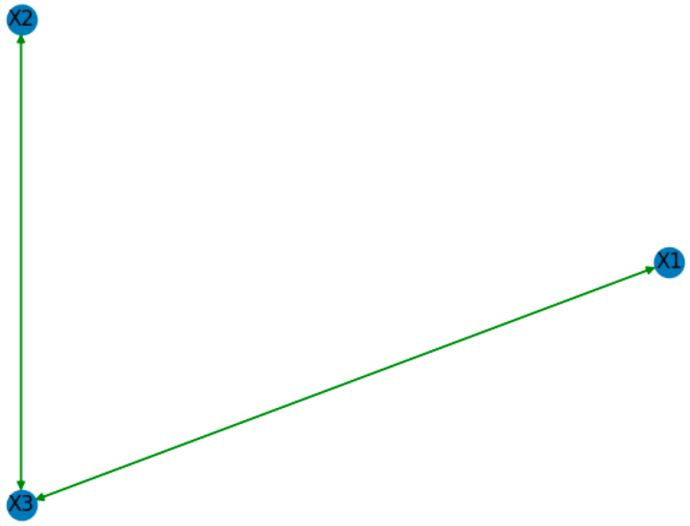

Figure 4: One of the graphs recovered by Peter-Clark algorithm on latents estimated by a non-factorised identifiable variational autoencoder. This is using synthetic data, and approximately recovers the relationship between the two latents (Z_1, Z_2 here appear as X_1, X_2) and the observed variable Y (here corresponding to X_3. This graph isn't quite correct: the edge from X_3 should be directed toward X_2, and the edge from X_1 to X_3. Generated using the causal-learn library [cla20].

4.3 Implementation

The implementation of the nf-iVAE is adapted from the source code[1] provided by the original authors of [KKMH20]. The network is implemented in Pytorch [PGM+19]. I used two different implementations of the Peter-Clark algorithm, one from the Causal Discovery Toolbox library [KG19] and the other from the causal-learn library [cla20]. The kernel conditional independence test is one of the options the PC algorithm can use to perform an independence test, and in both cases I use the KCI test implementation that ships with the library.

4.4 Performance metric

While the nf-iVAE maximises the evidence lower bound of the data (ELBO), in the original iVAE paper an additional metric is used: the mean correlation coefficient (MCC) between the true latents and the estimated latents [KKMH20]. A high MCC score means that the network has successfully recovered the correct sources of the data.

5 Results

On the synthetic data, the nf-iVAE seems to successfully recover the latents, with MCC scores above 0.9. When using these latents as inputs to the PC algorithm, the graphs that are generated differ from one trained network to another, even where the training conditions were identical. Sometimes these differences are not material: for example, Figure 4 and Figure 5 are equivalent, since the absolute position of a node does not matter at all. On the other hand, sometimes they vary widely enough that there are extra edges and self-loops (variables seemingly causing themselves) that do not respect the causal structure of the underlying model, for example as in Figure 6.

One additional issue is that the graphs contain edges that point in both directions, which is not useful at all from a causal perspective. What we'd like to see in this case is that Y is caused by Z_1 (graph has an edge $Z_1 \to Y$) and in turn causes Z_2 (graph has directed edge $Y \to Z_2$). Beyond the causal discovery perspective, this is needed by ICaRL to determine the parents of a variable such that

[1] https://github.com/siamakz/iVAE

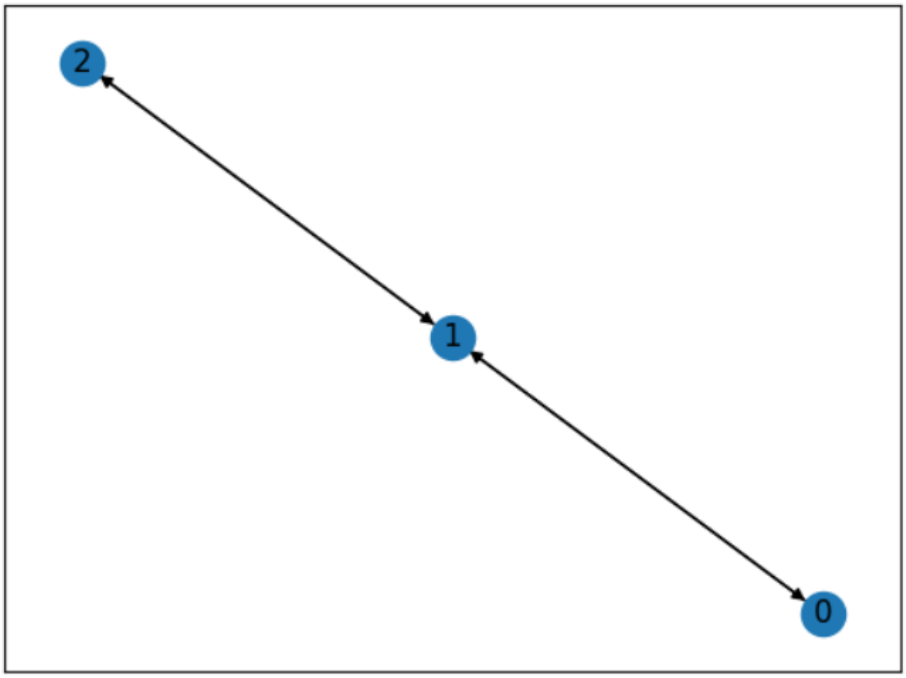

Figure 5: Another example of a graph recovered by the Peter-Clark algorithm based on the latents inferred by the nf-iVAE. This graph is equivalent to the previous one, and again does not capture the correct direction of causality between variables. Generted using the Causal Discovery Toolbox [KG19].

it can use this invariant representation of Y to learn an optimal classifier across environments. In other words, if it's not possible to recover the parents, ICaRL does not generalise.

The experiments on dSprites did not converge, with the nf-iVAE struggling to learn an approximate distribution over the latents. It's likely that this was due to an implementation error when adapting the networks implementing the VAE to take in the extra latent variable.

One possible limitation of ICaRL I noticed during the experiments was that the kernel conditional independence test is very slow for large inputs. In [ZPJS12], the method is found to have $\mathcal{O}(n^3)$ complexity. In practice this severely constrains the dimensionality of the latent space of the nf-iVAE and implicitly limits the number of sources we could reasonably disentangle. I can imagine complicated enough causal graphs that the method becomes infeasible (for example, for medical applications), but it's unclear to me just how big of a problem this is.

6 Further work

It would be interesting to see a performance comparison of end-to-end ICaRL for out-of-distribution generalisation against alternative approaches in a controlled benchmark like DomainBed [GLP20]. Another benchmark for OOD generalisation is presented in [SvKT+21], where the authors find that most common architectures do not generalise OOD in the visual domain. Ultimately, we'd like to be able to learn models that understand how data arises as a combination of sources. This type of model could have many useful properties: it would generalise robustly across environments, since causal factors seem to be broadly invariant across domains and it would also allow us to study counterfactuals, because we could arbitrarily modify recovered latents to explore what would happen in a situation that was not observed in the data.

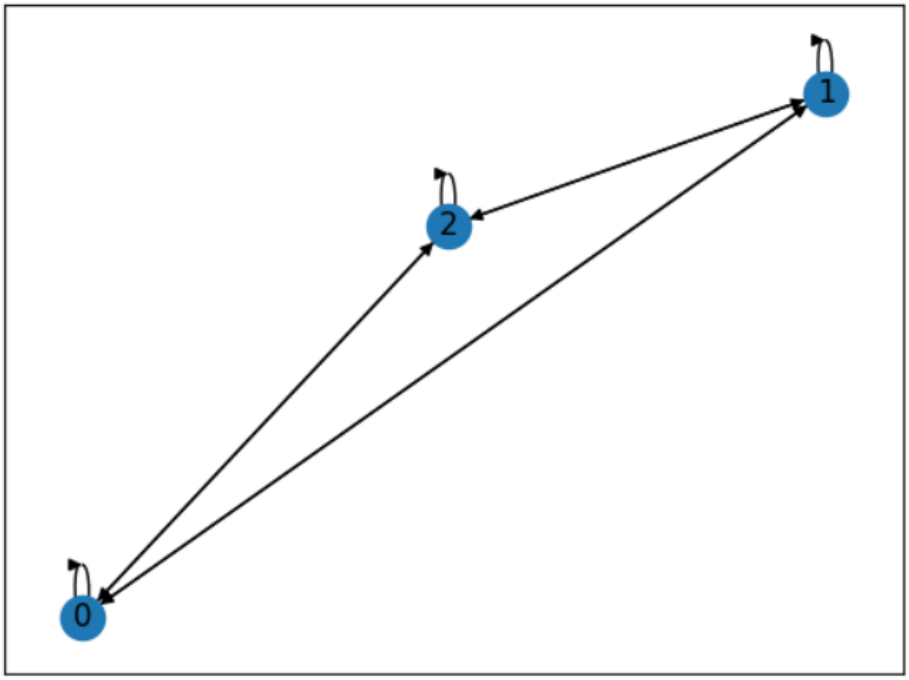

Figure 6: Example of a nonsensical graph sometimes output by the PC algorithm on the latent factors estimated by the nf-iVAE. This causal graph does not encode anything useful: every variable causes every variable, including itself.

References

[ABGLP19] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[Arj20] Martin Arjovsky. *Out of distribution generalization in machine learning*. PhD thesis, New York University, 2020.

[BVHP18] Sara Beery, Grant Van Horn, and Pietro Perona. Recognition in terra incognita. In *Proceedings of the European conference on computer vision (ECCV)*, pages 456–473, 2018.

[cla20] The causal-learn authors. Causal discovery for python. translation and extension of the tetrad java code. https://github.com/cmu-phil/causal-learn, 2020.

[Com94] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[GLP20] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020.

[HMP+16] 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. 2016.

[KG19] Diviyan Kalainathan and Olivier Goudet. Causal discovery toolbox: Uncover causal relationships in python. *arXiv preprint arXiv:1903.02278*, 2019.

[KKMH20] Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics*, pages 2207–2217. PMLR, 2020.

[KW13] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[LWHLS21] Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2021.

[MHHL17] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[PBM16] Jonas Peters, Peter Bühlmann, and Nicolai Meinshausen. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(5):947–1012, 2016.

[Pea09] Judea Pearl. Causal inference in statistics: An overview. *Statistics surveys*, 3:96–146, 2009.

[PGM+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[SGSH00] Peter Spirtes, Clark N Glymour, Richard Scheines, and David Heckerman. *Causation, prediction, and search*. MIT press, 2000.

[SvKT+21] Lukas Schott, Julius von Kügelgen, Frederik Träuble, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. *arXiv preprint arXiv:2107.08221*, 2021.

[ZPJS12] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Kernel-based conditional independence test and application in causal discovery. *arXiv preprint arXiv:1202.3775*, 2012.