Learning Latent Structural Causal Models From Low-level Data

Jithendaraa Subramanian

Computer Science
McGill University, Montreal

January 21, 2024

Acknowledgements

I would first like to thank my parents, to whom I am eternally grateful for nurturing my intellectual curiosity and encouraging me to follow my passion.

I am grateful to my supervisors Derek Nowrouzezahrai and Samira Ebrahimi Kahou for guiding me through my master's studies. Their trust in my capabilities and their support in my exploration of causal representation learning have been instrumental to my growth.

I would like to acknowledge and thank my peers, friends, and collaborators (in alphabetical order): *Aniket Didolkar, Anirudh Goyal, Cristian Dragos Manta, Dhanya Sridhar, Divyat Mahajan, Jason Hartford, Krishna Murthy Jatavallabhula, Kshitij Gupta, Mizu Nishikawa-Toomey, Moksh Jain, Nanda Harishankar Krishna, Nan Rosemary Ke, Nishka Katoch, Prishruit Punia, Sébastien Lachapelle, Shagun Sodhani, Shivakanth Sujit, Shruti Joshi, Soma Karthik, Stefan Bauer, Tristan Deleu, Vedant Shah, Yashas Annadani.* They made my masters really memorable!

I would also like to acknowledge some of the key suggestions that helped improve the problem formulation. Rosemary suggested using Gaussian intervention values which led to significantly improved experimental results. Tristan provided invaluable guidance regarding the math and model evaluation, which gave me clarity regarding the problem formulation. Anirudh's emphasis on the model's practical utility motivated me to conduct experiments on image generation from unseen interventions. Discussions with Sébastien and Dhanya made me pay more attention to causal identifiability.

Abstract

Causal learning primarily focuses on uncovering the concealed causal mechanisms that enhance our understanding of data beyond its usual distribution. Existing research in this field has commonly made the assumption that causal variables are predefined and observed. However, machine learning applications frequently witness learning from low-level data, such as image pixels or high-dimensional vectors. In these scenarios, the entire Structural Causal Model (SCM), comprising its structure, parameters, and high-level causal variables, remains latent and requires learning from the observed low-level data.

This thesis investigates the problem of Bayesian Inference Over Latent SCMs (BIOLS) from low-level data. BIOLS is introduced as a practical method for approximate inference, simultaneously inferring the latent SCM's causal variables, structure, and parameters from known interventions. To assess the effectiveness of BIOLS, experiments are conducted on synthetic datasets and a benchmark image dataset characterized by causal associations. Additionally, BIOLS's capability to generate images from previously unseen interventional distributions is demonstrated. These findings highlight the potential benefits of causal representation learning for generalization in downstream tasks.

Résumé

L'apprentissage causal se concentre principalement sur la découverte des mécanismes causaux cachés qui améliorent notre compréhension des données au-delà de leur distribution habituelle. Les recherches existantes dans ce domaine ont généralement fait l'hypothèse que les variables causales sont prédéfinies et observées. Cependant, les applications d'apprentissage automatique sont souvent confrontées à l'apprentissage à partir de données de bas niveau, telles que les pixels d'images ou les vecteurs de haute dimension. Dans ces scénarios, l'ensemble du Modèle Causal Structurel (SCM), comprenant sa structure, ses paramètres et ses variables causales de haut niveau, reste latent et nécessite un apprentissage à partir des données de bas niveau observées.

Cette thèse examine le problème de l'Inférence Bayésienne sur les SCMs Latents (BIOLS) à partir de données de bas niveau. BIOLS est présenté comme une méthode pratique pour l'inférence approximative, permettant simultanément d'inférer les variables causales latentes du SCM, sa structure et ses paramètres à partir d'interventions connues. Pour évaluer l'efficacité de BIOLS, des expériences sont menées sur des ensembles de données synthétiques et un ensemble de données d'images de référence caractérisé par des associations causales. De plus, la capacité de BIOLS à générer des images à partir de distributions d'interventions précédemment inconnues est démontrée. Ces résultats mettent en lumière les avantages potentiels de l'apprentissage de représentations causales pour la généralisation dans les tâches ultérieures.

Contents

List of Figures

List of Tables

1

Introduction

The question of whether machines can think and learn has long fascinated researchers across fields, with early efforts to explore this concept dating back to the 1940s (Hebb 1949; Turing 1950; Rosenblatt 1958). During these times, AI research focused on developing algorithms that could perform specific tasks such as playing chess or translating languages by manipulating symbols. This approach, known as symbolic AI, dominated the field for many years. However, symbolic AI systems often struggled to perform tasks that required complex pattern recognition or decision-making, leading to the development of alternative approaches.

In 1943, McCulloch and Pitts proposed the McCulloch-Pitts neuron (now known as the Perceptron) as a computational unit inspired by neurons in the brain (McCulloch and Pitts 1943). The Perceptron (Rosenblatt 1958), a machine designed for image classification, was the first implementation of this concept and initially consisted of a single layer linear classifier. Despite demonstrating signs of learning and intelligence, connectionist AI approaches such as the Perceptron were not seen as a viable option for simulating intelligence due to the dominance of symbolic AI at the time. The publication of "Perceptrons" by Minsky and Papert 1969 further hindered the advancement of connectionist models by highlighting their inability to learn complex, nonlinear functions, and even simpler functions such as the XOR.

In the decades that followed, advancement of computer hardware and the idea

of using the error backpropagation algorithm resurfaced, which made the training of deeper neural networks feasible. The error backpropagation algorithm was independently discovered many times since the 1960s (Kelley 1960; Dreyfus 1962), but saw widespread applications to train deep neural networks only after its popularisation in 1986 by David Rumelhart et al (Rumelhart, Hinton, and Williams 1986a; Rumelhart, Hinton, and Williams 1986b). Training deeper networks meant that artificial neural network could now learn more complex, nonlinear dependencies unlike the Perceptron.

Since then, deep learning has seen tremendous success in various applications. A flurry of neural network architectures have been proposed to perform feature extraction from various kinds of sensory inputs (images, videos, audio): Convolutional Neural Networks (LeCun et al. 1989; Fukushima 1980) to handle feature extraction from pixels, recurrent models (Hochreiter and Schmidhuber 1997) to handle temporal data for speech recognition and machine translation, Variational Autoencoders (Kingma and Welling 2013) and Generative Adversarial Networks (Goodfellow et al. 2014) to handle generation, and recently, the introduction of the attention mechanism (Bahdanau, Cho, and Y. Bengio 2014) and Transformers (Vaswani et al. 2017).

However, much of deep learning, and machine learning in general, is limited in many tasks that humans and animals excel at, such as transfer and generalization. The way in which AI fails is also very different from how natural intelligence fails. Many of these limitations arise due to statistical nature of the learning algorithms as we will discuss in the next section. Hence, the focus of this thesis is develop a method to learn causal relationships among high level variables from low-level data, and investigate how the learned causal relationships might be useful for overcoming the limitations of statistical learning and prediction.

1.1 Artificial Neural Networks

In this section, we will discuss fundamental concepts in deep learning, and in particular, ideas surrounding the use multiple layers of neurons with nonlinear activation functions to build universal function approximators for complex pattern recognition – a feat that the original work on Perceptrons fell short of.

1.1.1 Multilayer Perceptrons

As mentioned earlier, the Perceptron was a single layer classifier without any activation function. This severely restricted the function f, that transformed the inputs (x) to the output class (y), to a linear function class. This prevented the network from recognizing complex patterns, sometimes even simple ones such as the XOR. Multilayer Perceptrons (MLP) emerged to address this limitation. By stacking multiple layers of neurons and applying a nonlinear activation function after the linear transformation at each layer, the MLP was able to capture progressively more complex patterns in data and extract features. Many nonlinear activation functions have been proposed since, such as ReLU (Nair and Hinton 2010), Tanh, Sigmoid for binary classifications tasks, and the Softmax function for multi-class classification tasks.

This ability to model arbitrarily complex functions is very promising since it allows the objective $\mathcal{L}(y, f_\phi(x))$ to reach lower minima for any function $\mathcal{L}(.)$[1] given the right learning rule on how to update the MLP parameters, ϕ.

1.1.2 Learning with Gradient Descent

Now that we can perform a forward pass (by running the MLP on inputs and calculating outputs of each layer till the last) to predict a $\hat{y}$ arbitrarily close to y, all we need is an algorithm that can propose how to change the parameters ϕ so that the

[1] though in practice, we choose this to be a convex function such as the mean squared error for easier optimization

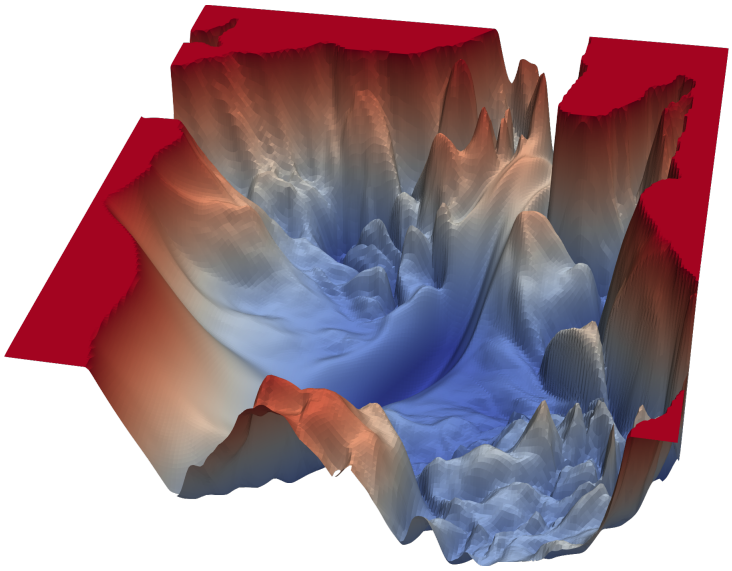

Figure 1.1: 3D visualization of a high-dimensional loss landscape from an example problem. Source image from here.

objective decreases. The gradient, $\nabla_\phi \mathcal{L}(y, f_\phi(x))$, is a local measure of how much the objective increases as a result of making an incremental change to the parameters ϕ, and thus the negative of this value gives a local measure of how much the objective decreases. The update rule for gradient descent is given by

$$\phi_{new} \leftarrow \phi_{old} - \alpha \nabla_\phi \mathcal{L}(y, f_\phi(x)) \ , \tag{1.1}$$

where α is the learning rate or step size (of how much we want to move in parameter space). By doing many such updates, one expects to converge to a local minima that is sufficiently close to the global minima in the loss landscape (figure 1.1). However, having a large step size can overshoot the minima and the learning might not converge. Lowering α can help in this case, but as $\alpha \to 0$ the time taken to converge tends to infinity. To help convergence and its speed, several variants of gradient descent have been proposed.

1.1.3 Escaping Local Minima: Variants of Gradient Descent

Stochastic Gradient Descent (Robbins 1951): The gradient $\nabla_\phi \mathcal{L}(y, f_\phi(x))$ often consists of a lot of terms since the loss is a summation over a large number of data

samples. The number of computations to be performed also grows with the number of parameters in the model. Hence, calculating the complete gradient at one go can be costly and even impossible when compute is limited.

Algorithm 1 Stochastic gradient descent (optionally with momentum)

Require: $\alpha, \phi_0, \mathcal{L}_B(\phi)$ (objective), μ (momentum), τ (dampening)
Ensure: ϕ_T
  for t=1 $\ldots T$ do **do**
    $g_t \leftarrow \nabla_\phi \mathcal{L}_B(\phi_{t-1})$
    if $\mu \neq 0$ **then**
      if $t > 1$ **then**
        $\mathbf{b_t} \leftarrow \mu \mathbf{b_{t-1}} + (1 - \tau) g_t$
      end if
    end if
    $g_t \leftarrow \mathbf{b_t}$
    $\phi_t \leftarrow \phi_{t-1} - \alpha g_t$
  end for

Mini-batch stochastic gradient descent *seeks to obtain an estimate of the gradient instead*, by randomly calculating the gradient, $\nabla_\phi \mathcal{L}_B(y, f_\phi(x))$, with respect to a mini-batch of samples. The stochasticity thereby introduced also allows the model to escape saddle points and sharp local minima and seek flatter local minima which tend to generalize better. When a single sample is used to get this gradient estimate, it is termed as stochastic gradient descent (i.e., batch size 1, and more noisy), and is called mini-batch gradient descent when the batch-size is larger than 1 (less noisy estimates for larger batch sizes). This can optionally be implemented with momentum, where the current update depends on the weighted average over previous gradient estimates. A sudden gradient change thus does not adversely affect the gradient trajectory.

Algorithm 1 summarizes stochastic gradient descent optionally with momentum. Other common optimization schemes such as **Adagrad** (Duchi, Hazan, and Singer 2011) and **Adam** (Kingma and Ba 2014) are summarized in algorithms 2 and 3.

Algorithm 2 Adagrad optimization

Require: $\alpha, \phi_0, \mathcal{L}, \eta$ (learning rate decay)
Ensure: ϕ_T
  $state_0 \leftarrow 0$
  for t=1 $\ldots T$ do **do**
    $g_t \leftarrow \nabla_\phi \mathcal{L}(y, f_{\phi_{t-1}}(x))$
    $\tilde{\alpha} \leftarrow \frac{\alpha}{1+(t-1)\eta}$
    $state_t \leftarrow state_{t-1} + g_t^2$
    $\phi_t \leftarrow \phi_{t-1} - \tilde{\alpha}\frac{g_t}{\sqrt{state_t}+\epsilon}$
  end for

Algorithm 3 Adam optimization

Require: $\alpha, \phi_0, \mathcal{L}, \beta_1, \beta_2$
Ensure: ϕ_T
  $m_0 \leftarrow 0$ (first moment)
  $v_0 \leftarrow 0$ (second moment)
  for t=1 $\ldots T$ do **do**
    $g_t \leftarrow \nabla_\phi \mathcal{L}(y, f_{\phi_{t-1}}(x))$
    $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$
    $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$
    $\widehat{m_t} \leftarrow \frac{m_t}{(1-\beta_1^t)}$
    $\widehat{v_t} \leftarrow \frac{v_t}{(1-\beta_2^t)}$
    $\phi_t \leftarrow \phi_{t-1} - \alpha\frac{\widehat{m_t}}{\sqrt{\widehat{v_t}}+\epsilon}$
  end for

1.1.4 The Backpropagation Algorithm

We have so far seen different variants for updating the parameters with the gradient, but how can one calculate the gradient of the objective with respect to every parameter in the network? For single layer networks, this calculation is quite straightforward. But this is not the case for deeper networks. The only straightforward calculation in this case is the gradient with respect to parameters of the last layer (say $\phi(n) \in \phi$). To obtain the gradient of the objective $\mathcal{L}(.)$, with respect to any parameter $\phi(i)$ in the i^{th} layer, one has to use the chain rule in accordance with the backpropagation algorithm (Kelley 1960; Rumelhart, Hinton, and Williams 1986b):

$$\frac{\partial \mathcal{L}(y, f_\phi(x))}{\partial \phi(i)} = \frac{\partial \mathcal{L}(y, f_\phi(x))}{\partial a(n)} \cdot \frac{\partial a(n)}{\partial a(n-1)} \cdots \frac{\partial a(i+1)}{\partial \phi(i)}\ , \tag{1.2}$$

where $a(k)$ is the activation after the k^{th} layer. The backprop algorithm thus provides a way to propagate errors in a local fashion – there is information on how to update each layer of parameters, given the gradient with respect to weights of the next layer in the deep neural network. There are other ways, however, to do this credit assignment such as Lee et al. 2014; Akrout et al. 2019 but currently, most alternatives do not work as well as backprop. Interestingly, despite its success, there is some evidence in neuroscience that backpropagation might not be biologically plausible due to the weight transport problem.

1.1.5 The Reparametrization Trick

The objective functions and activations used thus far in the learning framework seem to be deterministic functions. However, it is not clear to immediately see the application to probabilistic learning: backpropagrating through stochastic functions and sampling operations (Y. Bengio, Léonard, and A. C. Courville 2013). The reparametrization trick is a way of representing a sampling process in a way so that backpropagration is possible, and is very commonly used in Variational Autoencoders (Kingma and Welling 2013).

Consider the stochastic node in the computational graph of the forward pass, where a sample is drawn from a univariate Gaussian, $z \sim \mathcal{N}(\mu, \sigma^2)$. We wish to obtain the gradient with respect to parameters of the distribution: μ and σ. Backpropagating directly through the stochastic node is not possible, however, rewriting the sampling process as $z \coloneqq \mu + \sigma\epsilon$, where ϵ is drawn from a standard normal distribution allows backpropagation (figure 1.2). One can see this is equivalent since $\mathbb{E}[z] = \mathbb{E}[\mu + \sigma\epsilon] = \mathbb{E}[\mu] + \mathbb{E}[\sigma\epsilon] = \mu + \sigma\mathbb{E}[\epsilon] = \mu$. The variance can likewise be calculated to be σ^2 using $Var[z] = \mathbb{E}[z^2] - \mathbb{E}[z]^2$. This alternative representation can therefore allow one to estimate gradients, $\frac{\partial z}{\partial \mu}$ and $\frac{\partial z}{\partial \sigma}$. For sampling from a multivariate Gaussian $\mathcal{N}(\mu, \Sigma)$, a similar representation holds: $\boldsymbol{z} \coloneqq \boldsymbol{\mu} + L\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I)$ and L is the cholesky

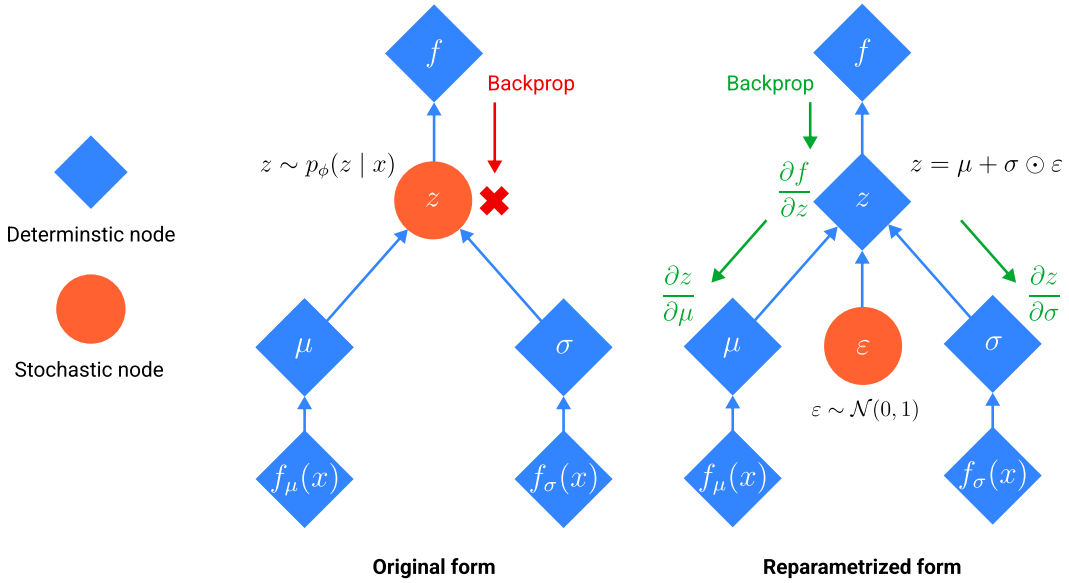

Figure 1.2: Reparametrization trick for sampling from a univariate Gaussian (figure inspired by these lecture slides)

decomposition of Σ.

Reparametrization tricks are not limited to continuous-valued distributions. For discrete distributions, a commonly used approach is to rewrite the discrete sampler as a continuous one with a temparature hyperparameter τ. The function is constructed in such a way that annealing the temperature $\tau \to 0$ induces a discrete behaviour, but is continuous in general.

Common examples include the Relaxed Bernoulli distribution and the Gumbel-Softmax distribution as a continuous approximator of the categorical distribution (Jang, Gu, and Poole 2016). Yet another example is sampling doubly stochastic matrices using the Sinkhorn operator (Sinkhorn 1964) and the Hungarian algorithm (Kuhn 1955) which has applications in permutation learning (Helmbold and Warmuth 2009; Diallo, Zopf, and Fürnkranz 2020) and causal discovery (Cundy, Grover, and Ermon 2021), as we will see in chapter 5.3.

1.2 Parameter Sharing as Inductive Bias

1.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN) (Fukushima 1980; LeCun et al. 1989) are built to handle image data and are particularly useful for tasks in vision such as image recognition, image compression, and image generation. Since spatial orientation is a distinguishing property of image data, the CNN learns features by applying the same function in different local areas of the image. In other words, the inductive bias of a CNN is to *share parameters across space*.

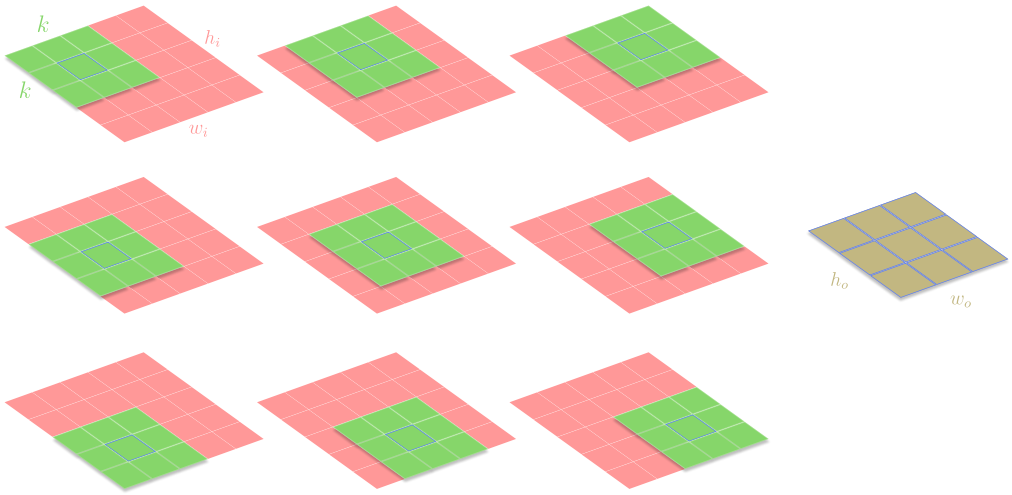

Figure 1.3: Image convolution with a stride of 1

The key computation in the CNN is a kernel filter (green) that performs local computation on each pixel in the input image (red) to produce feature maps (gold) to be used in the subsequent layers to capture more high-level features for the task at hand. The kernel moves over the image with a stride, row by row like a sliding window, and performs elementwise multiplication followed by a sum to obtain the pixel value at that position. Since this is a linear operation, capturing complex patterns requires

alternating layers of CNN and nonlinear activations. The relationship between input resolution (h_i, w_i), output resolution (h_o, w_o), kernel size (k, k), padding (p_h, p_w) and stride (s_h, s_w) is given by equations 1.3 and 1.4.

$$h_o = \left\lfloor \frac{h_i + 2 \cdot p_h - k}{s_h} + 1 \right\rfloor \tag{1.3}$$

$$w_o = \left\lfloor \frac{w_i + 2 \cdot p_w - k}{s_w} + 1 \right\rfloor \tag{1.4}$$

1.2.2 Recurrent Neural Networks

Recurrent networks are built to handle temporal data and are useful for a wide variety of tasks: neural machine translation, speech recognition, music generation, video extrapolation (handled by a recurrent models of CNN), weather forecasting, and stock market predictions, to name a few. The inductive bias of these models is to *share parameters across time*, and the model operates under the assumption that the same function is responsible for temporal transitions of a variable: $x(0) \rightarrow x(1) \ldots x(t-1) \rightarrow x(t)$.

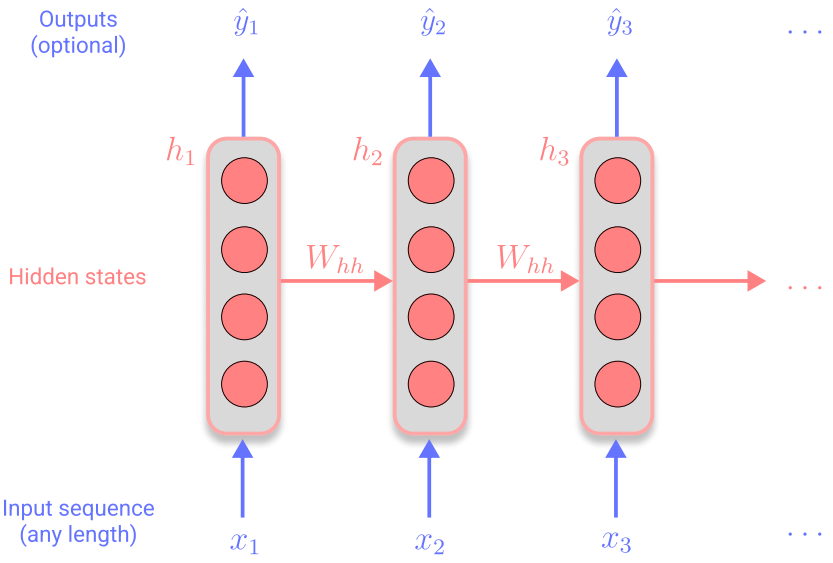

Figure 1.4: A recurrent neural network

Recurrent Neural Networks (RNN) use their internal state at timestep $t-1$ and an input to update their internal state and produce the output for the next timestep t. For each element in the input sequence, the RNN layer performs the following computation:

$$\boldsymbol{h_t} = \sigma(W_{ih}^T \boldsymbol{x_t} + W_{hh} \boldsymbol{h_{t-1}} + \boldsymbol{b}) \tag{1.5}$$

where $\sigma(.)$ is the sigmoid function, $\boldsymbol{h_t} \in \mathbb{R}^h$ refers to the hidden state at timestep t, $\boldsymbol{x}_t \in \mathbb{R}^i$ refers to the input vector at timestep t, $W_{ih} \in \mathbb{R}^{i \times h}$ is a weight matrix that embeds the input vector in hidden space, $W_{hh} \in \mathbb{R}^h$ is a weight matrix to perform a linear operation on the hidden state, and $\boldsymbol{b} \in \mathbb{R}^h$ is a bias vector. The initial hidden state $\boldsymbol{h}_0$ is typically initialized to zeros or a random vector, or is learned in some cases. Optionally, a weight matrix $W_{hi} \in \mathbb{R}^{h \times i}$ can be learnt to project the hidden vector back into input space. Using the predictions from previous timestep, an RNN can autoregressively unroll to arbitrarily many timesteps, independent of the length of the input or output sequence.

1.2.2.1 Exploding and Vanishing Gradient Problem

Consider the gradient of the loss at timestep i with respect to the hidden state h_j at some previous timestep j:

$$\frac{\partial \mathcal{L}_i(\theta)}{\partial \boldsymbol{h_j}} = \frac{\partial \mathcal{L}_i(\theta)}{\partial \boldsymbol{h_i}} \prod_{j<t\leq i} \frac{\partial \boldsymbol{h}_t}{\partial \boldsymbol{h}_{t-1}} \tag{1.6}$$

$$= \frac{\partial \mathcal{L}_i(\theta)}{\partial \boldsymbol{h_i}} W_{hh}^{(i-j)} \prod_{j<t\leq i} \operatorname{diag}\Big(\sigma'(W_{ih}^T \boldsymbol{x_t} + W_{hh} \boldsymbol{h_{t-1}} + \boldsymbol{b})\Big) \tag{1.7}$$

Taking the L2 matrix norm on both sides, and applying $||AB|| \leq ||A|| \cdot ||B||$ (L2 matrix norm is submultiplicative), we get:

$$\left\| \frac{\partial \mathcal{L}_i(\theta)}{\partial \boldsymbol{h_j}} \right\| \leq \left\| \frac{\partial \mathcal{L}_i(\theta)}{\partial \boldsymbol{h_i}} \right\| \cdot \left\| W_{hh}^{(i-j)} \right\| \prod_{j<t\leq i} \left\| \operatorname{diag}\Big(\sigma'(W_{ih}^T \boldsymbol{x_t} + W_{hh} \boldsymbol{h_{t-1}} + \boldsymbol{b})\Big) \right\| \tag{1.8}$$

In this way, Pascanu, Mikolov, and Y. Bengio 2013 showed that the gradient norm vanishes or explodes when backpropagating through long sequences, depending on the

largest eigenvalue of W_{hh} (explodes for > 1, and vanishes for < 1). For the exploding gradient problem, a common fix is gradient clipping. Vanishing gradients are harder to deal with and researchers have sought to other variants of RNN (and recently, to attention) such as Long Short-term Memory (LSTM) and Gated Recurrent Units (GRU). Residual connections also address the vanishing gradient problem.

1.2.2.2 Long Short Term Memory

LSTM (Hochreiter and Schmidhuber 1997; Sak, Senior, and Beaufays 2014) is a type of RNN that maintains a cell state c_t apart from the usual hidden state h_t, and was proposed as a way to alleviate the vanishing gradients problem. The cell state c_t, stores long-term information and acts as a "memory". The LSTM also has several gates that can erase, read, and write information from the cell state and operates according to the equations given below, where $\odot$ is the Hadamard product:

$$\text{[input gate]} \quad \boldsymbol{i}_t = \sigma(W_{ii}\boldsymbol{x}_t + W_{hi}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{hi}) \tag{1.9}$$

$$\text{[forget gate]} \quad \boldsymbol{f}_t = \sigma(W_{if}\boldsymbol{x}_t + W_{hf}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{hf}) \tag{1.10}$$

$$\text{[cell gate]} \quad \boldsymbol{g}_t = \tanh(W_{ig}\boldsymbol{x}_t + W_{hg}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{hg}) \tag{1.11}$$

$$\text{[output gate]} \quad \boldsymbol{o}_t = \sigma(W_{io}\boldsymbol{x}_t + W_{ho}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{ho}) \tag{1.12}$$

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{g}_t \tag{1.13}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t) \tag{1.14}$$

1.2.2.3 Gated Recurrent Units

The GRU (Cho et al. 2014) was proposed as a simpler alternative to the LSTM and has fewer parameters to learn, but also addresses the vanishing gradient problem. Notably, it no longer uses a cell state but rather uses: (i) **an update gate** that decides which parts of the hidden state are updated or preserved and, (ii) **a reset gate** that decides which parts of h_{t-1} are used to compute the new content. Like the LSTM, these mechanisms can retain long-term information, thus not having to rely

on gradients that are too far in the past. The mechanism of the GRU is driven by these equations given below, where $\odot$ is the Hadamard product:

$$\text{[reset gate]} \quad \boldsymbol{r}_t = \sigma(W_{ir}\boldsymbol{x}_t + W_{hr}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{hr}) \tag{1.15}$$

$$\text{[update gate]} \quad \boldsymbol{u}_t = \sigma(W_{iu}\boldsymbol{x}_t + W_{hu}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{hu}) \tag{1.16}$$

$$\text{[new gate]} \quad \boldsymbol{n}_t = \tanh(W_{in}\boldsymbol{x}_t + r_t \odot (W_{hn}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{hn}) + \boldsymbol{b}_{in}) \tag{1.17}$$

$$\boldsymbol{h}_t = (1 - \boldsymbol{u}_t) \odot \boldsymbol{n}_t + \boldsymbol{u}_t \odot \boldsymbol{h}_{t-1} \tag{1.18}$$

1.2.3 The Attention Mechanism

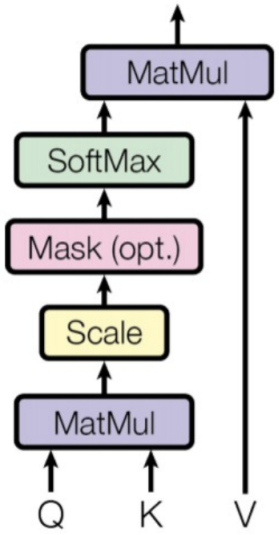

Figure 1.5: Operations for Dot Product Attention (from (Vaswani et al. 2017))

In cases where one has to refer to a set of hidden states in the past, a set of attention scores can be learnt, one for each of the hidden state. A (usually linear) combination of the attention scores and the hidden states can then be taken as a context vector for the task at hand. Intuitively, the attention mechanism can be thought of as learning residual connections to the past hidden states with different weights. This avoids long sequences and largely solves the vanishing gradient problem. The attention mechanism was first introduced in a task for neural machine translation (Bahdanau,

Cho, and Y. Bengio 2014) but has been extended to other tasks since then – for example, slot attention (Locatello et al. 2020) for images, and SAVi (Elsayed et al. 2022) for videos.

The most commonly used form of attention is the dot product attention (Vaswani et al. 2017), which consists of M keys $K \in \mathbb{R}^{M \times d_k}$, queries $Q \in \mathbb{R}^{d_q}$ and values $V \in \mathbb{R}^{M \times d_v}$. The keys and queries are projected to a common subspace in $\mathbb{R}^d$ via $K_d = \text{MLP}(K)$ and $Q_d = \text{MLP}(Q)$. The calculation of attention scores $\boldsymbol{\alpha} \in \mathbb{R}^M$ and the mechanism of attention is given below.

$$\boldsymbol{\alpha} = Softmax\Big(\frac{K_d Q_d}{\sqrt{d_k}}\Big) \tag{1.19}$$

$$Attention(K, Q, V) = \sum_{i=1}^{M} \alpha_i \boldsymbol{V}_i \tag{1.20}$$

1.3 Limitations of Deep Learning

Sole Reliance on Statistical Learning: It is well known that the foundations of current machine learning and deep learning are based on statistical learning techniques. While these techniques have proven effective for making predictions, they do not necessarily uncover the underlying mechanisms that give rise to the statistical dependencies. For example, the frequency of storks (X) is a good predictor of the human birth rate in Europe (Matthews 2000), say Y, but this is clearly not the *mechanism* that dictates human birth rate. A statistical learning approach would be perfectly fine with exploiting such spurious correlations, but if we were to move to Canada, this statistical dependency might not hold anymore and is therefore not a generalizable predictor of human birth rate. Confounding variables can often lead to these spurious correlations. To put it simply, a variable Z may have been the cause of X and Y in Europe and hence one can use X to predict Y. However, in Canada, Z might not be causing X anymore (say, due to an intervention), and thus X cannot

be used to predict Y.

The i.i.d Assumption: Another limitation of present-day learning approaches is the assumption of independent and identically distributed (i.i.d.) data. In a typical statistical learning problem, we are given multiple samples from the i.i.d joint distribution $P(X, Y)$[2], and we want to learn a predictor $P(Y \mid X)$. However, if the distribution changes during test time due to an intervention, the predictor may not perform well. This can have significant consequences, such as in the case of self-driving cars. In fact, some approaches rely so heavily on the i.i.d assumption that even a small perturbation to a single pixel can significantly alter the predictions of an image classification system (Su, Vargas, and Sakurai 2019). These adversarial examples are not restricted to just pixel data. To mitigate this issue, data augmentation techniques are often used. However, a more fundamental solution would be to allow the model to explicitly train and test on non-i.i.d data, allowing it to distinguish between cause variables and merely correlated variables. Relaxing the i.i.d assumption is key to achieving generalization.

The Problem of Transferring and Modularizing Knowledge Transfer and modularization of knowledge are important considerations in the use of neural networks and go hand-in-hand. However, neural networks are often treated as black boxes for specific tasks, making it challenging to reuse specific modules in novel scenarios. To illustrate the importance of modularization, consider a dataset with altitude (A) and temperature (T) data. The joint distribution can be represented in two ways: $P(A)P(T \mid A)$ or $P(T)P(A \mid T)$. The first factorization, corresponding to the causal factorization, consists of two disentangled modules: $P(A)$ and $P(T \mid A)$. It is well-known that altitude directly affects temperature, and the module $P(T \mid A)$ represents this general physical mechanism. This means that in a new location, the learned would only need to relearn the parameters for the altitude module $P(A)$, and could

[2]Here, X and Y are some arbitrary random variables such as image pixels and labels, not referring to our earlier example of storks and human birth rates.

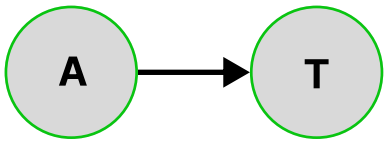

Figure 1.6: A causal diagram encoding the physical mechanism affecting altitude and temperature

continue using the parameters for the mechanism, $P(T \mid A)$.

In contrast, the second factorization (the anti-causal factorization), consists of two modules: $P(T)$ and $P(A \mid T)$. Since there is no generic mechanism by which altitude affects temperature, the module $P(A \mid T)$ cannot be reused, hindering transfer. As a result, the model would require more samples to learn in the new location. Therefore, proper modularization of knowledge is essential for effective transfer learning, but this aspect is often overlooked in current deep learning approaches.

1.4 A Potential Remedy

Causal learning may offer a solution to these limitations of statistical deep learning. The toolkit of causality (Pearl 2009b) and progress towards causal representation learning (Schölkopf, Locatello, et al. 2021a) offer potential ways to tackle these challenges and are also crucial for reinforcement learning. The ability to invent high-level concepts to explain low-level data and discover cause-effect relationships has the potential to enable the estimation of the effects of interventions and counterfactual reasoning, which is central for agent introspection and scientific discovery. This could allow for AI to transition from the intuitive system 1 cognition to the algorithmic and deliberate thinking that humans do (system 2 cognition) (Kahneman 2011).

1.5 Outline

This thesis studies the challenging tasks of causal discovery and learning latent causal models from low-level data (e.g. pixels). An exposition of representation learning, and related tools for inference and generation is provided in chapter 2. Fundamental concepts in causality are detailed in chapter 3 and serves as a prerequisite for much of the thesis. These include Structural Causal Models (SCM), the problem of causal discovery, and the task of latent causal discovery from low-level data. Chapter 4 presents a detailed discussion of research related to causal discovery and causal representation learning. A new algorithm for learning a joint distribution over high-level causal variables, causal structure, and mechanisms from low-level data (such as high-dimensional vectors and images) is proposed in chapter 5.3 by learning to generate data. The proposed approach is evaluated in the experiments presented in chapter 6. A discussion of limitations and directions for future work is given in chapter 7 before concluding the thesis.

2

Tools for Representation Learning

Density estimation is an important tool in statistical learning and is central to representation learning. It refers to estimating an unobserved probability density function (or probability mass function for discrete distributions) from observed data. If the class of density function is known or assumed to be known (e.g, Gaussian, Bernoulli), then only the distribution parameters have to be estimated. Though there are many methods to solve density estimation, common approaches include *Maximum Likelihood Estimation* (MLE), *Maximum A Posteriori* (MAP) estimation, and *Bayesian inference*.

2.1 Maximum Likelihood Estimation

Suppose that we collected a dataset of N observations $\mathcal{D} = (x^{(1)}, \ldots, x^{(N)})$ and want to explain the data using a random variable. The random variable can be assumed to come from an arbitrary density with distribution parameters θ (for a univariate Gaussian, this corresponds to μ and σ). MLE aims to find an optimal θ^*_{MLE} that maximizes the fit of observing exactly the dataset $\mathcal{D}$ that has been collected. Specifically, we want to calculate $\theta^*_{MLE} = \arg\max_\theta p(\mathcal{D} \mid \theta) = \arg\max_\theta \prod_{i=1}^{N} p(x^{(i)} \mid \theta)$[1]. Since this product over probabilities can cause underflow especially for large N, there is a

[1]Note that decomposing $p(\mathcal{D} \mid \theta)$ as $\prod_{i=1}^{N} p(x^{(i)} \mid \theta)$ only holds because of the i.i.d assumption

preference to operate in the log domain instead:

$$\theta^*_{MLE} = \arg\max_{\theta} \sum_{i=1}^{N} \log p(x^{(i)} \mid \theta) \tag{2.1}$$

As the name suggests, the optimization centers around maximizing the likelihood, $p(\mathcal{D} \mid \theta)$. For a Gaussian distribution, this is equivalent to minimizing $\frac{1}{2\sigma^2} \sum_{i=1}^{N} (x^{(i)} - \mu)^2 + \log \sigma$. If σ is taken to be constant, this corresponds to minimizing the commonly used Mean Squared Error (MSE).

2.2 Maximum A Posteriori Estimation

In Maximum A Posteriori (MAP) estimation, a prior is set on θ encoding domain knowledge (for example, setting a DAG-ness constraint or graph sparsity in structure learning) and the posterior is maximized instead. That is, $\theta^*_{MAP} = \arg\max_\theta p(\theta \mid \mathcal{D}) = \arg\max_\theta p(\mathcal{D} \mid \theta) p(\theta)$, since $p(\theta \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta) p(\theta)$. This is equivalent to:

$$\theta^*_{MAP} = \arg\max_{\theta} \log p(\theta) + \sum_{i=1}^{N} \log p(x^{(i)} \mid \theta) \tag{2.2}$$

Thus, when the prior is uniform (i.e., $\log p(\theta)$ is a constant), $\theta^*_{MAP} = \theta^*_{MLE}$. Even when this is not the case, it is easy to see that the MAP solution approaches MLE for large datasets since the likelihood term overpowers the prior.

2.3 Bayesian Inference

The MLE and MAP estimators previously discussed are point estimates since a single value for θ is obtained. However, there are cases where point estimates do not work well. For example, θ^*_{MAP} might not be representative of the whole posterior $p(\theta \mid \mathcal{D})$. In these scenarios, it might be desirable to obtain the entire posterior:

$$p(\theta \mid \mathcal{D}) = \frac{p(\mathcal{D} \mid \theta) p(\theta)}{\int_\theta p(\mathcal{D} \mid \theta) p(\theta) d\theta} \tag{2.3}$$

This is straightforward in cases where the prior and likelihood are conjugate and when the integral is tractable. However, there are many cases where the integral in the evidence $\int_\theta p(\mathcal{D} \mid \theta)p(\theta)d\theta$ in intractable. For discrete θ (in structure learning, this refers to different possible directed acyclic graphs), the evidence is a sum over an exponential number of terms, making this computation intractable. In such cases, one can resort to learning an approximate posterior instead of obtaining the exact posterior in closed form. In extreme cases, it might not even be possible to obtain an analytic expression for the posterior.

2.3.1 Variational Inference

In variational inference, the goal is to learn an approximate distribution $q_\phi(\mathbf{Z})$ over a set of unobserved variables $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$, given an i.i.d observed dataset $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, that is close to the true posterior, $p(\mathbf{Z} \mid \mathbf{X})$. In other words, we are interested in, minimizing between the two distributions:

$$D_{KL}(q_\phi(\mathbf{Z})||p(\mathbf{Z} \mid \mathbf{X})) = \mathbb{E}_{q_\phi}\left[\log \frac{q_\phi(\mathbf{Z})}{p(\mathbf{Z} \mid \mathbf{X})}\right] = \log p(\mathbf{X}) - \mathbb{E}_{q_\phi}\left[\log p(\mathbf{X} \mid \mathbf{Z}) - \log \frac{q_\phi(\mathbf{Z})}{p(\mathbf{Z})}\right] \tag{2.4}$$

Since the KL divergence is always positive, we obtain a lower bound on the log evidence:

$$\log p(\mathbf{X}) \geq \mathbb{E}_{q_\phi}\left[\log p(\mathbf{X} \mid \mathbf{Z}) - \log \frac{q_\phi(\mathbf{Z})}{p(\mathbf{Z})}\right] \tag{2.5}$$

Minimizing the KL divergence between the approximate and true posterior thus corresponds to maximizing this evidence lower bound (ELBO) $\mathcal{L}_\phi(q) = \mathbb{E}_{q_\phi}\Big[\log p(\mathbf{X} \mid \mathbf{Z}) - \log \frac{q_\phi(\mathbf{Z})}{p(\mathbf{Z})}\Big]$. In deep learning, we use an encoder network that maps inputs $\mathbf{X}$ to a distribution $q_\phi(\mathbf{Z})$. For the likelihood model, another network is used to map samples from $q_\phi(\mathbf{Z})$ to the observation space. The network is trained end-to-end by taking gradient ascent steps on the ELBO.

3

Causality

3.1 Structural Causal Models

The practice of using a Structural Causal Model to denote causal variables and their structure was introduced by Judea Pearl but has its roots in structural equation modeling and path analysis which has been studied extensively by geneticist Sewall Wright since 1918 (Wright 1918; Wright 1934).

A Structural Causal Model (SCM) (Pearl 2009b) over random variables $\mathbf{Z} \coloneqq \{Z_1 ....Z_d\}$ is defined by a set of equations which describe the mechanisms by which each endogenous variable Z_i depends on its direct causes $Z_{\pi_{\mathcal{G}}(i)}$ (parents of Z_i in graph $\mathcal{G}$) and an exogenous noise variable ϵ_i with probability density P_{ϵ_i}. If the causal parent assignment is assumed to be acyclic, then an SCM is associated with a Bayesian Network (BN) or a Directed Acyclic Graph (DAG) $\mathcal{G} = (V, E)$, where V corresponds to the endogenous variables and E encodes direct cause-effect relationships.

The exact value z_i taken on by the causal variable Z_i is given by local causal mechanisms f_i with parameters Θ_i that operate on the values of its parents $\mathbf{z}_{\pi_{\mathcal{G}}(i)}$ and the node's noise variable ϵ_i. This can be represented as:

$$z_i \coloneqq f_i(\mathbf{z}_{\pi_{\mathcal{G}}(i)}, \epsilon_i; \Theta_i) \ , \tag{3.1}$$

where $i = 1, \ldots, d$. Figure 3.1 illustrates an example SCM with a variable, its parents

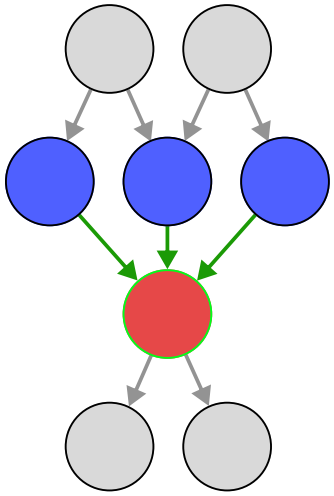

Figure 3.1: An example SCM with a highlighted variable (red), its parents (blue), and the causal mechanisms operating through cause-effect edges (green).

in the DAG, and the local causal mechanisms that operate via the edges.

For linear Gaussian SCMs with additive noise (given in equation 3.2), a class of SCMs that reoccurs in many parts of this thesis, all f_i's are linear functions, Θ_i reduces to $\boldsymbol{\theta}_i \in \mathbb{R}^d$ denoting the edge weights from every node to Z_i, and $\boldsymbol{A}_{\mathcal{G}_i} \in (0,1)^d$ refers to the i^{th} column of the adjacency matrix.

$$z_i \coloneqq \boldsymbol{\theta}_i^T (\boldsymbol{A}_{\mathcal{G}_i} \odot \mathbf{z}_{\pi_{\mathcal{G}}(i)}) + \epsilon_i \tag{3.2}$$

Here, $\mathbf{z}_{\pi_{\mathcal{G}}(i)} \in \mathbb{R}^d$ has the j^{th} element zeroed out whenever Z_j is not a parent of Z_i. The exogenous noise variables ϵ_i are taken to be a Gaussian random variable independent from each other, with 0 mean and variance σ_i^2. Alternatively, one can represent all the variables together in one equation:

$$\mathbf{Z} \coloneqq \theta^T \mathbf{Z} + \epsilon \ , \tag{3.3}$$

$$\mathbf{Z} \coloneqq (I - \theta)^{-T} \epsilon \ , \tag{3.4}$$

where $\mathbf{Z} \in \mathbb{R}^d$, $\theta \in \mathbb{R}^{d \times d}$ $\epsilon \in \mathbb{R}^d$ with mean of error variables $\mu_\epsilon = 0$ and covariance Σ_ϵ. It is worthwhile to note that one usually has assumptions of independent noise

variables since this is often a prerequisite for theoretical guarantees on identifying the true DAG or its Markov Equivalence Class (MEC) (Hoyer et al. 2008; Pearl 2009b; Peters, Mooij, et al. 2013).

It is a well known result that for linear Gaussian SCMs to be identifiable from data[1], one needs to operate under the assumption that all the error variables ϵ_i have the same variance (i.e., all $\sigma_i^2 = \sigma^2$) (Peters and Bühlmann 2012). The joint distribution over error variables is then an isotropic Gaussian with covariance $\Sigma_\epsilon = \sigma^2 I_d$, where I_d is the identity matrix of dimension d.

An SCM thus entails a joint distribution over the d random variables:

$$\mathbf{Z} \sim \mathcal{N}(0, \Sigma_{\mathbf{Z}}) \quad \text{where} \quad \Sigma_{\mathbf{Z}} = (I - \theta)^{-T} \Sigma_\epsilon (I - \theta)^{-1} \tag{3.5}$$

that obeys the causal factorization according to $\mathcal{G}$,

$$p(Z_1, \ldots Z_d) = \prod_{i=1}^{d} p(Z_i \mid Z_{\pi_{\mathcal{G}}(i)}) \ . \tag{3.6}$$

3.2 Traditional Assumptions

Causal Markov Assumption: This states that any variable Z_i in the DAG G is independent of every other variable (except descendants of Z_i) conditional on all of its direct causes (Hausman and Woodward 1999).

The Faithfulness Assumption: If all the independence relations are a consequence of the Causal Markov condition, then we are said to be faithful to the DAG $\mathcal{G}$ (Spirtes, C. Glymour, and Scheines 2000). This is easier to see in a counterexample setting where one is unfaithful. Consider the three variable case: *Smoking, Exercise, Health*. *Exercise* has a positive impact on *Health*, while *Smoking* has a detrimental effect. However, people who smoke might be more health-conscious, making them want to exercise more often. In this way, *Smoking* has a positive impact on *Exercise*

[1] Here, data refers to samples of the causal variables.

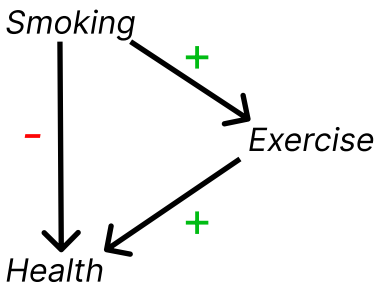

Figure 3.2: Example of a causal graph to illustrate the causal faithfulness assumption

and can indirectly improve *Health*. This structure could produce a wide array of distributions, depending on the local causal mechanisms (includes the functions as well as parameters). In some of these distributions, it is possible that the direct negative impact of *Smoking* on *Health* can balance the indirect positive effect of *Smoking* (through *Exercise*). Thus, it might appear that *Smoking* and *Health* do not have a cause-effect relationship, and that they are independent. In such a case, we say that one is unfaithful to the structure in figure 3.2. The Causal Faithfulness Assumption rules out independence statements that do not arise from the Causal Markov condition (d-separation).

Causal Sufficiency: This states that the set of measured variables (or variables that appear in the DAG $\mathcal{G}$), $\{Z_1 \dots Z_d\}$ in our case, includes the common causes of pairs of every (Z_i, Z_j). If this were not the case, one would have an implicit structure and dependency between some pairs (Z_i, Z_j) induced by some unmeasured variable U_{ij}.

3.3 Structure Learning and Causal Discovery

Structure learning in prior work refers to learning a DAG according to some optimization criterion with or without the notion of causality (e.g., He et al. 2019). The

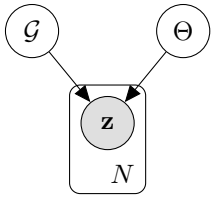

Figure 3.3: Bayesian Network for prior works in causal discovery and structure learning

task of causal discovery on the other hand, is more specific to learning the structure $\mathcal{G}$ (optionally also the mechanisms Θ) of SCMs, and subscribes to causality and interventions like that of Pearl 2009a. These approaches often resort to modular likelihood scores over causal variables – like the BGe (Geiger and Heckerman 1994; Kuipers, Suter, and Moffa 2022) or BDe (Heckerman, Geiger, and Chickering 1995) score – to learn the right structure. Furthermore, these approaches either obtain a maximum likelihood estimate,

$$\begin{aligned} \mathcal{G}^* &= \arg\max_{\mathcal{G}} p(Z \mid \mathcal{G}) \text{ or} \\ (\mathcal{G}^*, \Theta^*) &= \arg\max_{\mathcal{G},\Theta} p(Z \mid \mathcal{G}, \Theta) \end{aligned} \tag{3.7}$$

or in the case of Bayesian causal discovery (Heckerman, Meek, and Cooper 1997), variational inference is used to learn a joint posterior distribution $q_\phi(\mathcal{G}, \Theta)$ that approximates the true posterior $p(\mathcal{G}, \Theta \mid Z)$ by minimizing the KL divergence between the two

$$\arg\min_{\phi} D_{\mathrm{KL}}(q_\phi(\mathcal{G}, \Theta) \,||\, p(\mathcal{G}, \Theta \mid Z)) = \arg\max_{\phi} \mathbb{E}_{(\mathcal{G},\Theta)\sim q_\phi}\left[\log p(Z \mid \mathcal{G}, \Theta) - \log \frac{q_\phi(\mathcal{G}, \Theta)}{p(\mathcal{G}, \Theta)}\right] \tag{3.8}$$

where $p(\mathcal{G}, \Theta)$ is a prior over SCM structure and parameters possibly encoding DAG-ness or sparsity. Figure 3.3 shows the Bayesian Network (BN) over which inference is performed for causal discovery tasks.

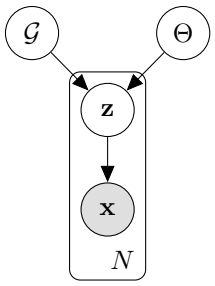

Figure 3.4: Bayesian Network for the latent causal discovery task that generalizes standard causal discovery setups

3.4 Latent Causal Discovery

In more realistic scenarios, one does not directly observe the raw causal variables – they must be learned from low-level data. The causal variables, structure, and parameters are associated with a latent SCM. The goal of causal representation learning models is to be able to: (i) infer the latent SCM from low-level data and, (ii) generate low-level samples from the inferred or learned latent SCM. Yang et al. 2021 proposes a Causal VAE but is in a supervised setup where one has labels on causal variables and the focus is on disentanglement. Kocaoglu et al. 2018 present causal generative models trained in an adversarial manner but assumes direct observations of the causal variables. Given the right causal structure as a prior, the work focuses on generation from conditional and interventional distributions.

In both the causal representation learning and causal generative model scenarios mentioned above, the Ground Truth (GT) causal graph and parameters of the latent SCM are arbitrarily defined on real datasets and the setting is supervised. Contrary to this, the approach we will study in this thesis is unsupervised or mildly-supervised (because of known intervention targets). This work is about recovering the ground truth underlying SCM and causal variables that generate the low-level observed data – we define this as the problem of *latent causal discovery*, and the Bayesian network over which we want to perform inference on is given in figure 3.4. We will have a detailed discussion of related work in the next chapter, before diving into the proposed

approach in chapter 5.3 and experimental results in chapter 6.

4

Related Work

Prior work can be classified into Bayesian (Koivisto and Sood 2004; Heckerman, Meek, and Cooper 2006; Friedman and Koller 2013) or maximum likelihood (Brouillard et al. 2020; Wei, Gao, and Y. Yu 2020; Ng, Zhu, et al. 2022) methods, that learn the structure and parameters of SCMs using either score-based (Kass and Raftery 1995; Barron, Rissanen, and B. Yu 1998; Heckerman, Geiger, and Chickering 1995) or constraint-based (Cheng et al. 2002; Lehmann and Romano 2005) approaches.

4.1 Causal Discovery

Within the realm of causal discovery, a significant body of work has emerged, primarily operating under the assumption that causal variables are directly observable and not derived from low-level data. Key contributions in this category include methods such as PC (Spirtes, C. N. Glymour, et al. 2000), Gadget (Viinikka et al. 2020), DAG-nocurl (Y. Yu, Gao, et al. 2021), and Z. Zhang et al. 2022. Chickering 2002 proposes a greedy search algorithm, but does not scale to a large number of nodes. Notably, Peters and Bühlmann 2014 provides a foundational insight by proving the identifiability of linear Gaussian Structural Causal Models (SCMs) with equal noise variances.

In the pursuit of causal discovery, various research directions have been explored. Y.

Bengio, Deleu, et al. 2019 employ the speed of adaptation as a signal for learning causal directions, while Ke, Bilaniuk, et al. 2019 focuses on learning causal models from unknown interventions. Further extending this exploration, Scherrer, Bilaniuk, et al. 2021; Tigas et al. 2022; Agrawal et al. 2019; Toth et al. 2022 resort to active learning and for causal discovery, focusing on performing targeted interventions to efficiently learn the structure.Ke, Chiappa, et al. 2022 proposed CSIvA, an approach that uses transformers (Vaswani et al. 2017) to learn causal structure from synthetic datasets and then generalize to more complex, naturalistic graphs. Ke, Dunn, et al. 2023 is a follow-up work to apply this to the problem of learning the structure of gene-regulatory networks.

Zheng et al. 2018 introduce an acyclicity constraint that penalizes cyclic graphs, thereby restricting search close to the DAG space. Lachapelle et al. 2019 leverages this constraint to learn DAGs in nonlinear SCMs. Building upon this constraint, Lachapelle et al. 2019 leverage it to learn DAGs within nonlinear SCMs. Temporal aspects of causal relationships are also explored, with methods like Pamfil et al. 2020 and Lippe, Magliacane, et al. 2022 specializing in structure learning from temporal data.

ENCO (Lippe, Cohen, and Gavves 2022) uses a score-based approach without acyclicity constraints to alternate between a "graph-fitting" phase and a "distribution fitting" phase to learn the structure of graphs. The method operates on observational and interventional data, and under mild conditions has convergence guarantees to obtain a DAG without the using common constraint based optimization techniques. It scales to up to 1000 nodes while having less than one mistake on average out of 1 million possible edges.

Other efforts include (Shimizu et al. 2011; Lopez-Paz and Oquab 2016; Y. Yu,

Chen, et al. 2019; Ghoshal and Honorio 2018; Ng, Ghassami, and K. Zhang 2020; Li et al. 2022).

Bayesian causal discovery: Annadani et al. 2021 casts the Bayesian structure learning problem as an autoregressive one by sequentially predicting edges, in hopes of capturing the potentially multi-modal posterior. Deleu, G'ois, et al. 2022 uses Generative Flow Networks, or GFlowNets (E. Bengio et al. 2021), a new class of probabilistic methods that lies at the intersection of reinforcement learning and variational inference. The work uses the transitive closure property ensuring that the action space is constrained to actions that do not introduce cycles. Nishikawa-Toomey et al. 2023 extends this to joint inference over structure and parameters using Variational Bayes (VB).Deleu, Nishikawa-Toomey, et al. 2023 is a closely related extension, however, instead of using a VB-based alternate optimization, a single GFlowNet is trained to satisfy the sub-trajectory balance condition, thus being able to sample a posterior over structures and parameters.

B. Wang, Wicker, and Kwiatkowska 2022 leverages sum product networks to perform exact Bayesian structure learning. Hägele et al. 2022 extends the framework of Lorch et al. 2021 to perform Bayesian causal discovery in a setting where interventions are unknown.

Table 4.1: Situating BIOLS in the context of related work in causal discovery.

	Joint G & θ	Unsupervised Z	Nonlinear SCM	Learn from low-level data
VCN (Annadani et al. 2021)	✗	✗	✗	✗
DiBS (Lorch et al. 2021)	✓	✗	✓	✗
DAG-GFN (Deleu, G'ois, et al. 2022)	✗	✗	✗	✗
VBG (Nishikawa-Toomey et al. 2023)	✓	✗	✗	✗
JSP-GFN (Deleu, Nishikawa-Toomey, et al. 2023)	✓	✗	✓	✗
BIOLS	✓	✓	✗	✓

Table 4.1 compares BIOLS with prior work in Bayesian Causal Discovery.

4.2 Latent Variables with Structure

Structure learning with latent variables: Markham and Grosse-Wentrup 2020 introduces the concept of Measurement Dependence Inducing Latent Causal Models (MCM). The proposed algorithm finds a minimal-MCM that induces the statistical dependencies between observed variables. However, similar to VAEs, the method assumes no causal links between latent variables. Kivva et al. 2021 provides the conditions under which the number of latent variables and structure can be uniquely identified for discrete latent variables, given the adjacency matrix between the hidden and measurement variables has linearly independent columns. Elidan et al. 2000 detects the signature of hidden variables using semi-cliques and then performs structure learning using the structural-EM algorithm (Friedman 1998) for discrete random variables.

Anandkumar et al. 2012 and Silva et al. 2006 consider the identifiability of linear Bayesian Networks when some variables are unobserved. However, it is important to note that the identifiability results in the former work are contingent upon specific structural constraints within the DAGs involved. Xie, Cai, et al. 2020 proposes the Generalized Independent Noise (GIN) condition to identify the structure between latent confounders, under the assumption of non-Gaussian noise and that certain sets of latents have a lower bound on the number of pure measurement child variables. Xie, Huang, et al. 2022 is in a setting where the edges exist not just between the latent variables but amongst low-level variables in the dataset as well. The frameworks discussed in the preceding works involve SCMs with a mix of observed and unobserved variables, whereas this thesis considers the entirety of the SCM as latent. Lastly, GraphVAE (He et al. 2019) learns a structure between latent variables but does not incorporate notions of causality such as interventions.

Latent variable models with predefined structure: Examples include the VAE (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014) which has an independence assumption between latent variables. To overcome this, Sønderby et al. 2016 and Zhao, Song, and Ermon 2017 define latent variables with a chain structure in VAEs. Kingma, Salimans, et al. 2016 uses inverse autoregressive flows to improve upon the diagonal covariance of latent variables in VAEs.

The formulation in all the above works involves *latent variable models with a predefined structure* or learning the structure of an SCM among *partially observed variables*. In contrast, our setup involves learning a structure among variables that are *completely unobserved*. That is, the entire SCM is latent in our setup.

4.3 Causal Representation Learning

Brehmer et al. 2022 present identifiability theory for learning causal representations from data pairs $(x, \tilde{x})$ before and after intervention on a single node, assuming fixed noise generated by the SCM. Ahuja, Hartford, and Y. Bengio 2022 studies identifiability in a related setup under sparse perturbations. Ahuja, Y. Wang, et al. 2022 discusses identifiability for causal representation learning when one has access to interventional data.

Other works (Kocaoglu et al. 2018; Shen et al. 2022; Moraffah et al. 2020) introduce generative models that use an SCM-based prior in latent space. In Shen et al. 2022, the goal is to learn causally disentangled variables. Yang et al. 2021 learn a DAG but needs labels of the (discrete) causal variables. Lopez-Paz, Nishihara, et al. 2017 establishes observable causal footprints in images by trying to learn the causal direction between every pair of variables.

Table 4.2 situates BIOLS amidst related work in causal representation learning and generative causal models.

Table 4.2: Situating BIOLS in the context of related work in causal generative models and causal representation learning.

	Joint G & θ	Learns any DAG	Unsupervised Z	Scaling nodes	Cont. Z	No constraints on paired data	Multi-target interventions
(Kocaoglu et al. 2018)	✗	✗	✗	•	✗	✓	•
(Yang et al. 2021)	✓	✓	✗	4	•	✓	•
(Shen et al. 2022)	✗	✓	✗	•	✓	✓	✗
(Brehmer et al. 2022)	✓	✓	✓	$8-10$	✓	✗	✗
BIOLS	✓	✓	✓	50+	✓	✓	✓

5

Learning Latent Structural Causal Models

The ability to learn causal variables and the dependencies between them is a crucial skill for intelligent systems, which can enable systems to make informed predictions and reasoned decisions, even in scenarios that diverge substantially from those encountered in the training distribution (Schölkopf, Locatello, et al. 2021b; Goyal and Y. Bengio 2022). In the context of causal inference, a Structural Causal Model (SCM) (Pearl 2009a) with a structure $\mathcal{G}$ and a set of mechanisms parameterized by Θ, induces a joint distribution $p(Z_1, \dots Z_d)$ over a set of causal variables. However, the appeal of SCM-based modeling lies in its capacity to represent a family of joint distributions, each indexed by specific interventions. Models can then be trained on samples from a subset of these joint distributions and can *generalize* to completely unseen joint distributions as a result of new interventions.

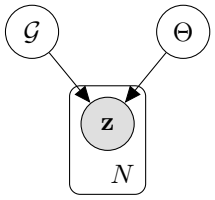

Figure 5.1: Bayesian Network for prior works in causal discovery and structure learning

Existing work on causal discovery aims to infer the structure and mechanisms of SCMs from observed causal variables (see Fig. 5.1). Learning such a causal model can then be useful for a wide-variety of downstream tasks like generalizing to out-of-

distribution data (Scherrer, Goyal, et al. 2022; Ke, Didolkar, et al. 2021), estimating the effect of interventions (Pearl 2009a; Schölkopf, Locatello, et al. 2021b), disentangling underlying factors of variation (Y. Bengio, A. Courville, and Vincent 2013; Y. Wang and Jordan 2021), and transfer learning (Schölkopf, Janzing, et al. 2012; Y. Bengio, Deleu, et al. 2019).

In high-dimensional problems typically studied in machine learning, neither the causal variables nor the causal structure relating them are known. Instead, the causal variables, structure and mechanisms have to be learned from high-dimensional observations, such as images (see Fig. 5.2).

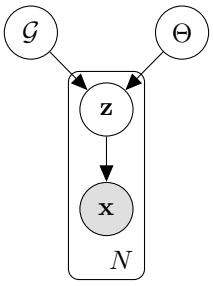

Figure 5.2: BN for the latent causal discovery task that generalizes standard causal discovery setups

An application of interest is in the context of biology, where researchers are interested in understanding Gene Regulatory Networks (GRN). In such problems, the genes themselves are latent but can be intervened on, the results of which manifest as changes in the high-resolution images (Fay et al. 2023). Here, the number of latent variables (genes) is known but the structure, mechanisms, and the image generating function remain to be uncovered. This serves as the motivation for our work.

Particularly, this thesis address the problem of inferring the latent SCM – including the causal variables $\mathcal{Z}$, structure $\mathcal{G}$ and parameters Θ – by learning a generative model of the observed high-dimensional data. Since these problems potentially have to be tackled in the low-data and/or non-identifiable regimes, this work adopts a Bayesian formulation so as to model the epistemic uncertainty over latent SCMs. Concretely, given a dataset of high-dimensional observations, the proposed approach BIOLS –

Bayesian Inference Over Latent SCMs – uses variational inference to model the joint posterior over the causal variables, structure and parameters of the latent SCM. Our contributions are as follows:

- We *propose a general algorithm, BIOLS, for Bayesian causal discovery in the latent space of a generative model*, learning a joint distribution over causal variables, structure and parameters in linear Gaussian latent SCMs with known interventions. Figure 5.3 illustrates an overview of the proposed method.
- By learning the structure and parameters of a latent SCM, BIOLS implicitly induces a joint distribution over the causal variables. Sampling from this distribution is equivalent to ancestral sampling through the latent SCM. As such, *we address a challenging, simultaneous optimization problem* that is often encountered during causal discovery in latent space: one cannot find the right graph without the right causal variables, and vice versa (Brehmer et al. 2022).
- On synthetically generated datasets and a benchmark image dataset (Ke, Didolkar, et al. 2021) called the chemistry environment, BIOLS consistently outperforms baselines and uncovers causal variables, structure, and parameters. We also demonstrate the ability of BIOLS to generate images from unseen interventional distributions.

5.1 Problem Setup

We are presented with a dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)}\}$, where each $\mathbf{x}^{(i)}$ represents high-dimensional observed data. For simplicity, we assume that $\mathbf{x}^{(i)}$ is a vector in $\mathbb{R}^{\mathrm{D}}$, but this setup can be extended to accommodate other types of inputs as well. Within this dataset, we posit the existence of latent causal variables $\mathbf{Z} = \{\mathbf{z}^{(i)} \in \mathbb{R}^d\}_{i=1}^N$, where $d \leq \mathrm{D}$, which underlie and explain the observed data $\mathcal{D}$. These latent variables

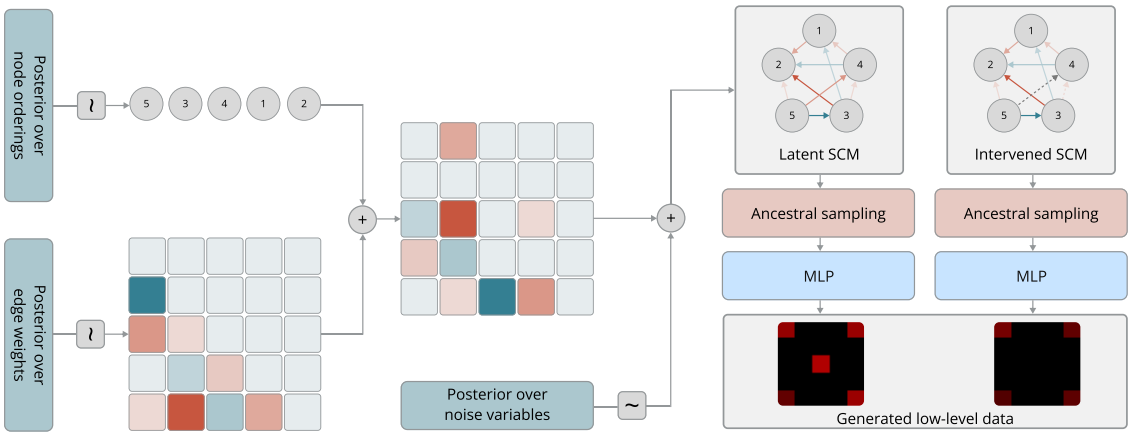

Figure 5.3: Model architecture of the proposed generative model for the Bayesian latent causal discovery task to learn latent SCMs from low-level data.

belong to a Ground Truth (GT) SCM, denoted by its structure $\mathcal{G}_{GT}$ and parameters Θ_{GT}. We wish to invert the data generation process $g : (\mathcal{G}_{GT}, \Theta_{GT}) \to \mathbf{Z} \to \mathcal{D}$. In the setting, we also have access to the intervention targets $\mathcal{I} = \{\mathcal{I}^{(i)}\}_{i=1}^{N}$ where each $\mathcal{I}^{(i)} \in \{0, 1\}^d$. The j^{th} dimension of $\mathcal{I}^{(i)}$ takes a value of 1 if node j was intervened on in row entry i, and 0 otherwise. To formalize the setup, we consider $\mathcal{X}$, $\mathcal{Z}$, $\mathcal{G}$, and Θ to represent the random variables over low-level data, latent causal variables, the SCM structure, and SCM parameters, respectively.

5.2 BIOLS: Bayesian Inference Over Latent SCMs

Here, we aim to estimate the joint posterior distribution $p(\mathcal{Z}, \mathcal{G}, \Theta \mid \mathcal{D})$ over the entire latent SCM. Computing the true posterior analytically requires calculating the marginal likelihood $p(\mathcal{D})$ which gets quickly intractable due to summation over the number of possible DAGs which grows super-exponentially with respect to the number of nodes. Thus, we resort to variational inference (Blei, Kucukelbir, and McAuliffe 2017) that provides a tractable way to learn an approximate posterior $q_\phi(\mathcal{Z}, \mathcal{G}, \Theta)$ with

variational parameters ϕ, close to the true posterior $p(\mathcal{Z}, \mathcal{G}, \Theta \mid \mathcal{D})$ by maximizing the Evidence Lower Bound (ELBO),

$$\mathcal{L}(\psi, \phi) = \mathbb{E}_{q_\phi(\mathcal{Z},\mathcal{G},\Theta)}\left[\log p_\psi(\mathcal{D} \mid \mathcal{Z}, \mathcal{G}, \Theta) - \log \frac{q_\phi(\mathcal{Z}, \mathcal{G}, \Theta)}{p(\mathcal{Z}, \mathcal{G}, \Theta)}\right], \tag{5.1}$$

where $p(\mathcal{Z}, \mathcal{G}, \Theta)$ is the prior distribution over the SCM, $p_\psi(\mathcal{D} \mid \mathcal{Z}, \mathcal{G}, \Theta)$ is the likelihood model with parameters ψ, mapping the latent causal variables to the observed high-dimensional data. An approach to learn this posterior could be to factorize it as

$$q_\phi(\mathcal{Z}, \mathcal{G}, \Theta) = q_\phi(\mathcal{Z}) \cdot q_\phi(\mathcal{G}, \Theta \mid \mathcal{Z}) \tag{5.2}$$

Given a procedure to estimate $q_\phi(\mathcal{Z})$, the conditional $q_\phi(\mathcal{G}, \Theta \mid \mathcal{Z})$ can be obtained using existing Bayesian structure learning methods (Cundy, Grover, and Ermon 2021; Deleu, Nishikawa-Toomey, et al. 2023). Otherwise, one has to perform a hard simultaneous optimization which would require alternating optimizations on $\mathcal{Z}$ and on $(\mathcal{G}, \Theta)$ given an estimate of $\mathcal{Z}$. Difficulty of such an alternate optimization is highlighted in Brehmer et al. 2022.

Alternate factorization of the posterior: Rather than decomposing the joint distribution as in equation 5.2, we propose to introduce a variational distribution $q_\phi(\mathcal{G}, \Theta)$ over only structures and parameters, so that the approximation of the joint posterior is given by $q_\phi(\mathcal{Z}, \mathcal{G}, \Theta) \approx p(\mathcal{Z} \mid \mathcal{G}, \Theta) \cdot q_\phi(\mathcal{G}, \Theta)$. The advantage of this factorization is that the true distribution $p(\mathcal{Z} \mid \mathcal{G}, \Theta)$ over $\mathcal{Z}$ is completely determined from the SCM given $(\mathcal{G}, \Theta)$ and exogenous noise variables (assumed to be Gaussian). This conveniently avoids the hard simultaneous optimization problem mentioned above since optimizing for $q_\phi(\mathcal{Z})$ is avoided. Hence, equation 5.1 simplifies to:

$$\mathcal{L}(\psi, \phi) = \mathbb{E}_{q_\phi(\mathcal{Z},\mathcal{G},\Theta)}\left[\log p_\psi(\mathcal{D} \mid \mathcal{Z}) - \log \frac{q_\phi(\mathcal{G}, \Theta)}{p(\mathcal{G}, \Theta)} - \cancelto{0}{\log \frac{p(\mathcal{Z} \mid \mathcal{G}, \Theta)}{p(\mathcal{Z} \mid \mathcal{G}, \Theta)}}\right] \tag{5.3}$$

Such a posterior can be used to obtain an SCM by sampling $\hat{\mathcal{G}}$ and $\hat{\Theta}$ from the approximated posterior. As long as the samples $\hat{\mathcal{G}}$ are always acyclic, one can perform ancestral sampling through the SCM to obtain predictions of the causal variables $\hat{\mathbf{z}}^{(i)}$. For additive noise models like in equation 3.6, these samples are already reparameterized and differentiable with respect to their parameters. The predictions of causal variables can then e bfed to the likelihood model to predict samples $\hat{\mathbf{x}}^{(i)}$ that best reconstruct the observed data $\mathbf{x}^{(i)}$.

5.3 Posterior parameterizations and priors

For linear Gaussian latent SCMs, which is the focus of this work, learning a posterior over $(\mathcal{G}, \Theta)$ is equivalent to learning $q_\phi(W, \Sigma)$, where W refers to weighted adjacency matrices W and Σ refers to covariance of $p(\epsilon)$, the distribution over noise variables of the SCM with 0 means. Supposing L to be the family of all adjacency matrices over a fixed node ordering, W and Σ parameterize the entire space of SCMs. Since $q_\phi(\mathcal{G}, \Theta) \equiv q_\phi(L, \Sigma)$, equation 5.3 leads to the following ELBO which has to be maximized, and the overall method is summarized in algorithm 4,

$$\mathcal{L}(\psi, \phi) = \mathbb{E}_{q_\phi(L,\Sigma)}\left[\mathbb{E}_{q_\phi(\mathcal{Z}|L,\Sigma)}[\log p_\psi(\mathcal{D} \mid \mathcal{Z})] - \log \frac{q_\phi(L,\Sigma)}{p(L)p(\Sigma)}\right] \tag{9}$$

Distribution over (L, Σ): The posterior distribution $q_\phi(L, \Sigma)$ has $\left(\frac{d(d+1)}{2}\right)$ elements to be learnt, and is parameterized by a diagonal covariance normal distribution. For the prior $p(L)$ over the edge weights, we promote sparse DAGs by using a horseshoe prior (Carvalho, Polson, and Scott 2009), similar to BCD Nets (Cundy, Grover, and Ermon 2021). A Gaussian prior is defined over $\log \Sigma$.

Algorithm 4 Bayesian latent causal discovery to learn $\mathcal{G}$, Θ, $\mathcal{Z}$ from high dimensional data

Require: $\mathcal{D}, \mathcal{I}$
Ensure: Approximate posterior distribution over $\mathcal{G}$, Θ, $\mathcal{Z}$
Initialize $q_\phi(L, \Sigma)$, $p_\psi(\mathcal{X} \mid \mathcal{Z})$, τ and set learning rate α
for num_epochs **do**
    $(\widehat{L}, \widehat{\Sigma}) \sim q_\phi(L, \Sigma)$
    $\widehat{W} \leftarrow \widehat{L}$
    for $i \leftarrow 1$ to N **do**
        $\mathcal{C}^{(i)} \leftarrow \text{argwhere}(\mathcal{I}^{(i)} = 1)$
        $\widetilde{W} = \text{copy}(\widehat{W})$
        $\widetilde{W}[:, \mathcal{C}^{(i)}] \leftarrow 0$ ▷ Mutated weighted adjacency matrix according to $\mathcal{I}^{(i)}$
        $\widehat{W}_{\mathcal{I}^{(i)}} \leftarrow \widetilde{W}$
        $\hat{\mathbf{z}}^{(i)} \leftarrow \text{AncestralSample}(\widehat{W}_{\mathcal{I}^{(i)}}, \widehat{\Sigma})$
    end for
    $\hat{\mathbf{Z}} \leftarrow \{\hat{\mathbf{z}}^{(i)}\}_{i=1}^{N}$
    $\hat{\mathcal{D}} \sim p_\psi(\mathcal{X} \mid \mathcal{Z} = \hat{\mathbf{Z}})$
    $\psi \leftarrow \psi + \alpha \cdot \nabla_\psi(\mathcal{L}(\psi, \phi))$ ▷ Update network parameters
    $\phi \leftarrow \phi + \alpha \cdot \nabla_\phi(\mathcal{L}(\psi, \phi))$
end for
return $\text{binary}(\widehat{W}), (\widehat{W}, \widehat{\Sigma}), \hat{\mathbf{Z}}$

6

Experimental Findings

In this chapter, we present experiments that evaluate the learned posterior over the linear Gaussian latent SCM. We aim to highlight the performance of our proposed method on latent causal discovery. As proper evaluation in such a setting would require access to the ground truth causal graph that generated the high-dimensional observations, we test our method against baselines on synthetically generated vector data and also on a benchmark dataset called the chemistry environment (Ke, Didolkar, et al. 2021) that causally generates images. Towards the end of the chapter, we evaluate the ability of our model to generate images from unseen interventional distributions.

6.1 Baselines

Since this is one of the early works to propose a working algorithm for leanring latent SCMs from high-dimensional data, there are currently no baseline methods that solve this task. However, we compare our approach against 4 baselines – **VAE**, **GraphVAE**, **ILCM** and **ILCM-GT**. While VAE has a marginal independence assumption between latent variables, GraphVAE (He et al. 2019) learns a DAG structure over latent variables. The final two baselines are ILCM (as introduced in Brehmer et al. 2022) and ILCM-GT, a variant of ILCM that directly uses the ground truth inter-

ventions instead of having to infer them. We include ILCM-GT to promote a fair comparison with BIOLS, since BIOLS does not infer interventions. It is to be noted that both VAE and GraphVAE are not designed to handle learning from interventional data. Additionally, while ILCM requires interventional data to train, the method also requires paired data inputs before and after an intervention with fixed noise over the unintervened nodes. For all baselines, we evaluate the quality of structure proposed by the learned model.

Evaluation metrics: We use two metrics commonly used in the literature – the expected Structural Hamming Distance (**$\mathbb{E}$-SHD**, lower is better) obtains the SHD (number of edge flips, removals, or additions) between the predicted and GT graph and then takes an expectation over SHDs of posterior DAG samples, and the Area Under the Receiver Operating Characteristic curve (**AUROC**, higher is better) where a score of 0.5 corresponds to a random DAG baseline. All our implementations are in JAX (Bradbury et al. 2018) and results are presented over 5 random DAGs.

In the following two paragraphs, we discuss the data generation procedure.

Generating the SCM: Following many works in the literature, we sample random Erdős–Rényi (ER) DAGs (Erdos, Rényi, et al. 1960) with degrees in $\{1, 2\}$ to generate the DAG. For every edge in this DAG, we sample the magnitude of edge weights uniformly as $|L| \sim \mathcal{U}(0.5, 2.0)$. Each of the d SCM noise variables is sampled as $\epsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, where $\sigma_i \sim \mathcal{U}(1, e^2)$.

Generating the causal variables and intervention targets: We then sample 20 random intervention sets where each set is a boolean vector denoting the intervention targets. An example of an intervention set for a 5-node DAG would be $[1, 0, 0, 1, 0]$. For each of these intervention sets, we generate 100 pairs of causal variables $(z_i, \tilde{z}_i)$ via ancestral sampling, where the intervention value is sampled from $\mathcal{N}(0, 2^2)$ with intervention noise added from $\mathcal{N}(0, 0.1^2)$. For nodes that were not intervened on, the same exogenous noise used to generate z_i is used as in Brehmer et al. 2022.

We now present 4 experimental setups to evaluate BIOLS where each experiment differs on how the generated causal variables are projected to $\mathcal{X}$.

$SO(n)$ **projection**: A random $SO(n)$ transformation $\mathbb{R}^d \to \mathbb{R}^d$ is made on the generated causal variables to obtain pairs $(x, \tilde{x})$, as done in Brehmer et al. 2022. Concretely, we take samples from $\mathcal{N}(0, 0.05)$ and fill these in the upper triangle of a $\mathbb{R}^{d \times d}$ zero matrix. The lower triangle is filled with the negative values such that the matrix is skew-symmetric. A linear projection is then performed with the matrix exponent of the skew-symmetric matrix to generate the transformed vectors. Following the linear projection, noise is sampled from $\mathcal{N}(\mathbf{0}, 0.1^2 * \mathbf{I_d})$, where $\mathbf{I_d}$ is the identity matrix of size d.

Linear projection: A random projection matrix of shape $\mathbb{R}^{d \times D}$ is initialized with each entry of the matrix sampled from $\mathcal{U}(-5, 5)$. Following the linear projection, noise is sampled from $\mathcal{N}(\mathbf{0}, 0.1^2 * \mathbf{I_D})$, where $\mathbf{I_D}$ is the identity matrix of size D. In our experiments, we set D to be 100.

Nonlinear projection: In more realistic scenarios, the ground truth generating function from $\mathcal{Z} \to \mathcal{X}$ is not just noisy but is also nonlinear. Thus we initialize a random 3-layer neural network to take care of the projection from d to D dimensions. In this setting, D is again set to 100. Noise is added in the same manner as the previous setting.

Chemistry environment: For this setting, intervention values are sampled from a standard Normal distribution instead. Once pairs of observational and interventional causal variables are generated, we use the chemistry environment to generate $(50, 50, 1)$ shaped images (Ke, Didolkar, et al. 2021), wherein there are d blocks with their intensity proportional to the d causal variables. In order to maintain stochasticity in the ground truth image generation process, we add noise sampled from $\mathcal{N}(0, 0.05^2)$ to the normalized (from 0 to 1) pixels of the image and bring them back to the $0 - 255$ range. Figure 6.1 shows an example image generated from the chemistry environment.

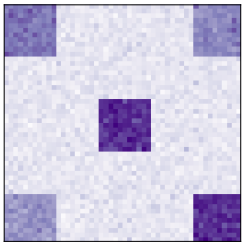

Figure 6.1: Image generated from the chemistry environment.

6.2 Results

In Figure 6.2, we provide a comprehensive summary of the results derived from BIOLS within the context of the initial experimental configuration, which pertains to a 5 node latent SCM and incorporates an $SO(n)$ transformation.

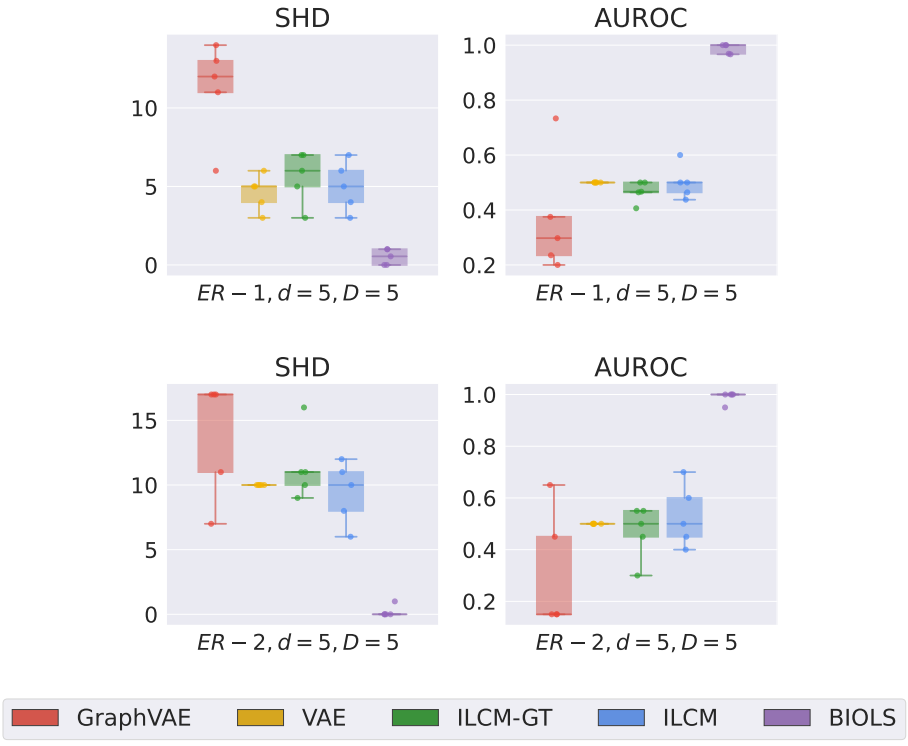

Figure 6.2: Learning 5−node SCMs of different graph densities (ER-1 and ER-2) from a 100−dimensional vector, where the generative function from $\mathcal{Z}$ to $\mathcal{X}$ is an $SO(n)$ transformation. $\mathbb{E}$**-SHD** (↓), **AUROC** (↑)

Figures 6.3 and 6.4 provide additional insights into the effectiveness of BIOLS, as they present the Expected Structural Hamming Distance (SHD) and the Area Under the Receiver Operating Characteristic (AUROC) for the acquired models in scenarios where the underlying generative function adopts two distinct forms: (i) a linear function and (ii) an arbitrary nonlinear function. It is worth noting that in both cases, the latent causal variables are projected to a 100-dimensional space.

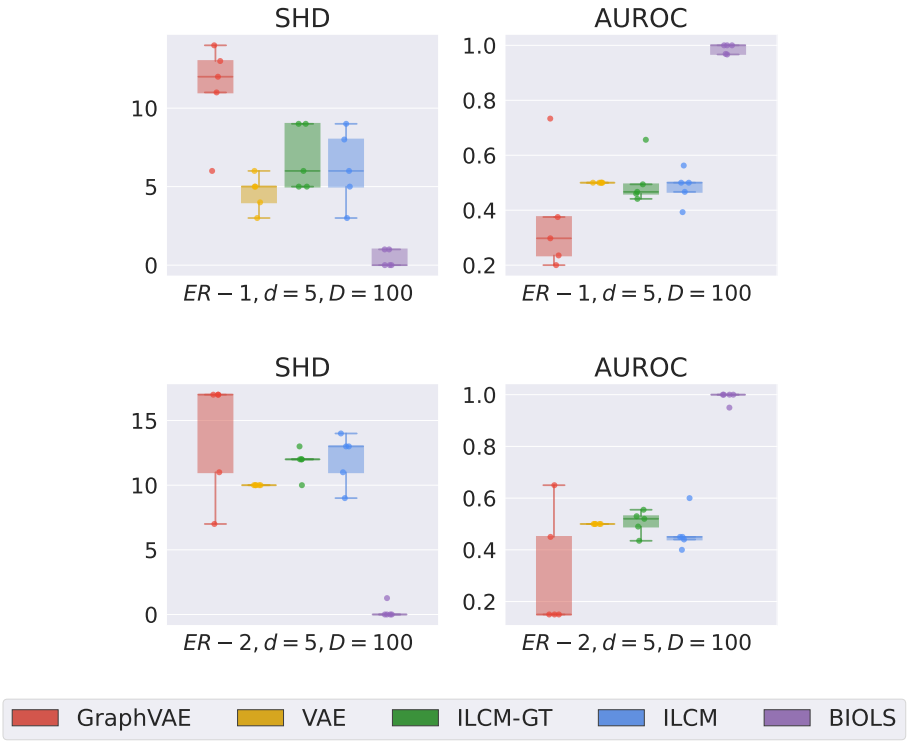

Figure 6.3: Learning 5−node SCMs of different graph densities (ER-1 and ER-2) from a 100−dimensional vector, where the generative function from $\mathcal{Z}$ to $\mathcal{X}$ is a linear projection to 100 dimensions. $\mathbb{E}$**-SHD** (↓), **AUROC** (↑)

Furthermore, we conducted experiments to assess the capacity of learning the latent SCM from image pixels. The summarized results are presented in Figure 6.5. Our findings demonstrate that BIOLS consistently outperforms baseline methods across all projection scenarios, including the intricate case of image data, yielding notable improvements in both evaluation metrics.

However, it is important to note that due to certain complexities in running ILCM

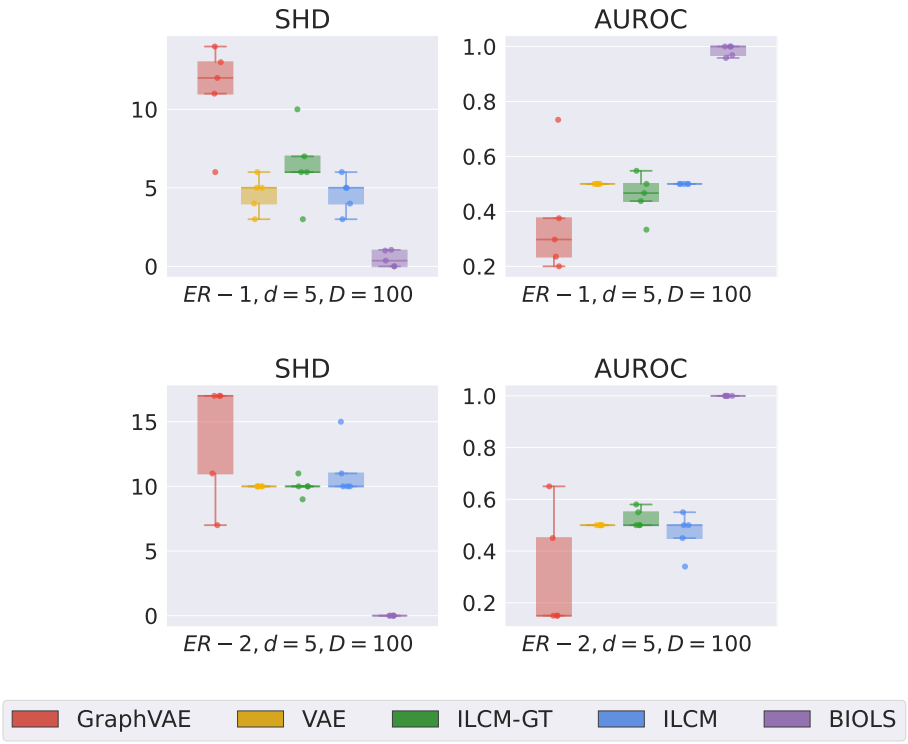

Figure 6.4: Learning 5−node SCMs of different graph densities (ER-1 and ER-2) from a 100−dimensional vector, where the generative function from $\mathcal{Z}$ to $\mathcal{X}$ is a nonlinear projection to 100 dimensions. $\mathbb{E}$**-SHD** (↓), **AUROC** (↑)

on image datasets, as outlined in the official implementation, we omit ILCM and ILCM-GT from the final set of image experiments.

Figure 6.6 showcases an evaluation of the model's capability to generate images from previously unseen interventional distributions within the chemistry dataset. This assessment is conducted by comparing the generated images with ground truth interventional samples. Notably, we observe a pattern in which the presence of a faint-colored or missing block in the first row corresponds to a light-colored block in the second row. This correspondence is reflective of matching causal variables, emphasizing the model's ability to generalize effectively. In summary, *we note that BIOLS consistently outperforms baselines, while maintaining a low SHD between* $0-1$.

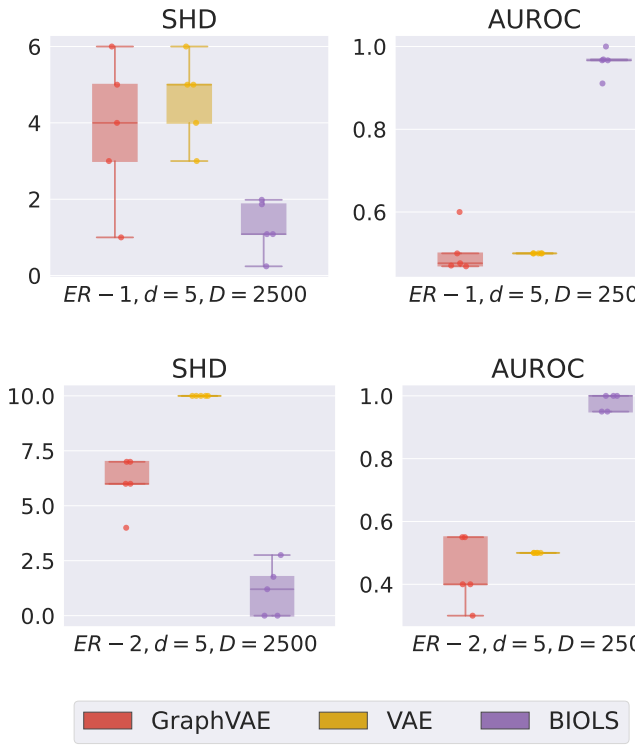

Figure 6.5: Learning 5−node SCMs of different graph densities (ER-1 and ER-2) from 50×50 images in the chemistry benchmark dataset (Ke, Didolkar, et al. 2021). $\mathbb{E}$**-SHD** (↓), **AUROC** (↑)

(a) Ground truth images sampled from 5 unseen interventional distributions.

(b) Images generated from 5 unseen interventional distributions using BIOLS.

Figure 6.6: Samples of images from the ground truth and learned interventional distributions. Intensity of each block refers to the causal variable. One block is intervened in each column.

6.2.1 Ablation on graph density

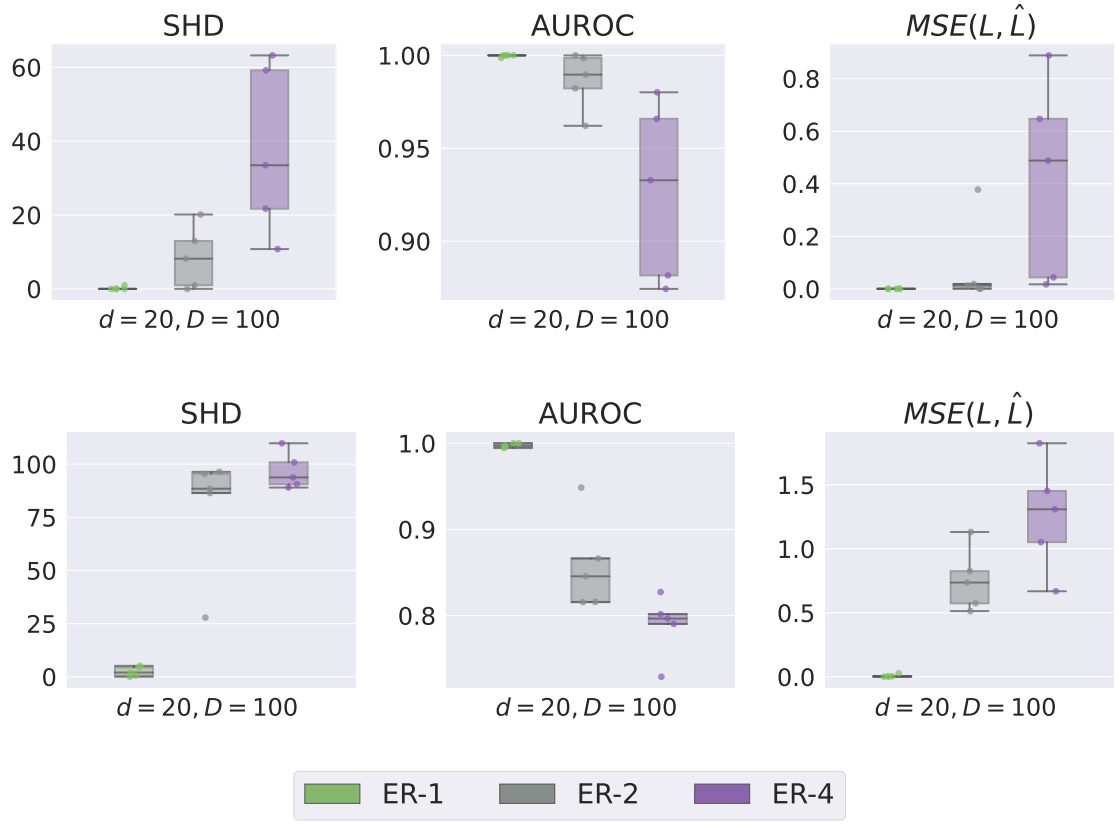

Figure 6.7: Effect of number of intervention sets on latent SCM recovery for linear (top row) and nonlinear (bottom row) generation function, $d = 20$ nodes. **SHD** $\downarrow$, **AUROC** $\uparrow$, **MSE**$(L, \hat{L}) \downarrow$

In this section, we perform ablations to study how graph density affects the quality of the SCM learned by BIOLS. Similar to other works in Bayesian structure learning, we notice that as the graph gets more dense, it gets harder to recover the SCM. Figure 6.7 illustrates the performance of BIOLS across the 3 metrics on $ER - 1$, $ER - 2$, $ER - 4$ graphs. These studies are on $d = 20$ node DAGs, projected to $D = 100$ dimensions. The model is trained on 120 intervention sets, with 100 samples per set to stay consistent with rest of the experiments. The top row in the figure corresponds to a linear projection between latent and observed variables. Similarly, the bottom row corresponds to the nonlinear projection.

Notably, we observe a trend wherein recovering edges becomes more challenging with denser graphs. This difficulty may arise from BIOLS needing to uncover a greater number of cause-effect relationships. This observation aligns with insights often noted in traditional causal discovery algorithms (e.g., Fig 5 and 12 in Scherrer, Bilaniuk, et al. 2021). It is important to note that the performance for denser graphs can be further improved by providing more interventional data (see section 6.2.2), increasing the variance of the Gaussian intervention values (refer section 6.2.4), or both.

6.2.2 Ablation on the number of intervention sets

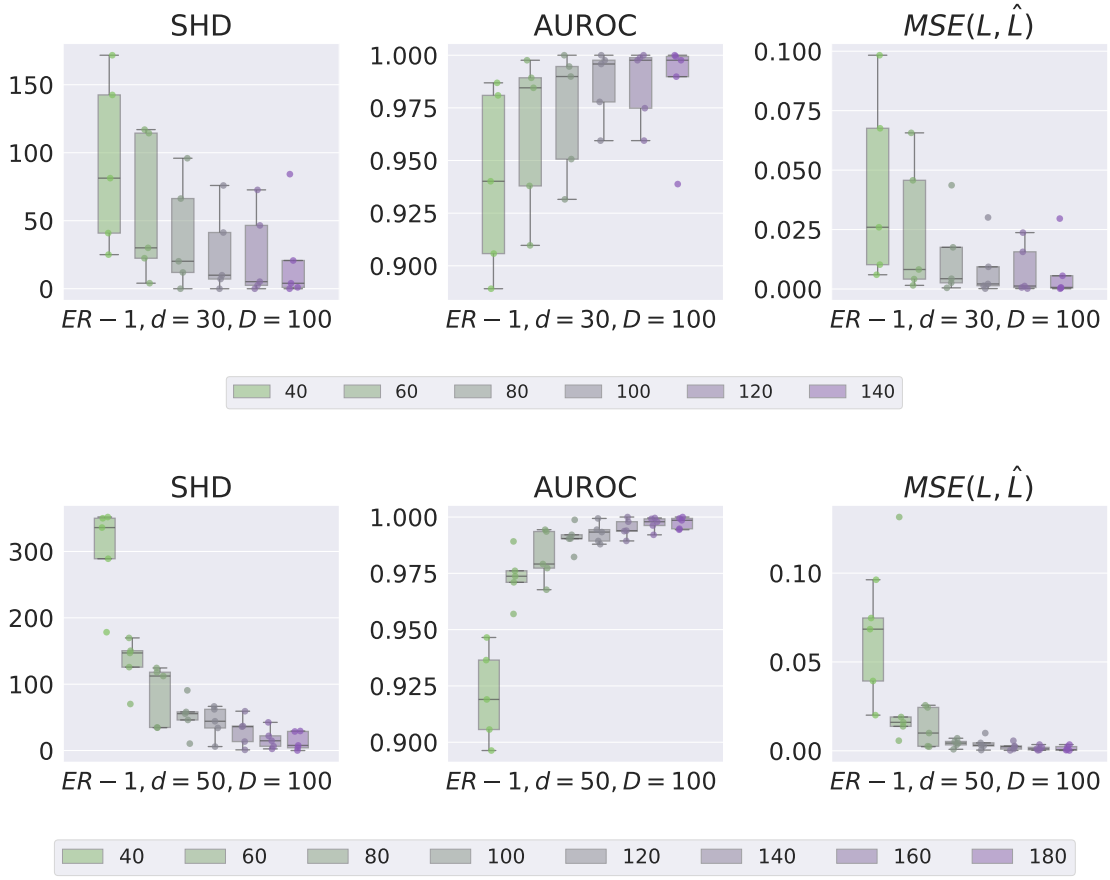

Figure 6.8: Effect of number of intervention sets on latent SCM recovery for a linear generation function, $d = 30, 50$ nodes. **SHD** $\downarrow$, **AUROC** $\uparrow$, **MSE**$(L, \hat{L})$ $\downarrow$

In the data generation phase, we saw that the interventional data is specified via two terms – number of intervention sets and number of interventional samples per set. In this subsection, we present some plots on how learning the latent SCM is affected for various number of nodes while varying the number of intervention sets. The number of interventional samples per set is kept constant and is set to 100 as in previous experiments.

Figure 6.8 demonstrates the effect of number of intervention sets on the quality of the latent SCM recovered by BIOLS, for a linear projection to $D = 100$ for 30 and 50 node SCMs. Figure 6.9 illustrates a similar plot for a nonlinear projection function.

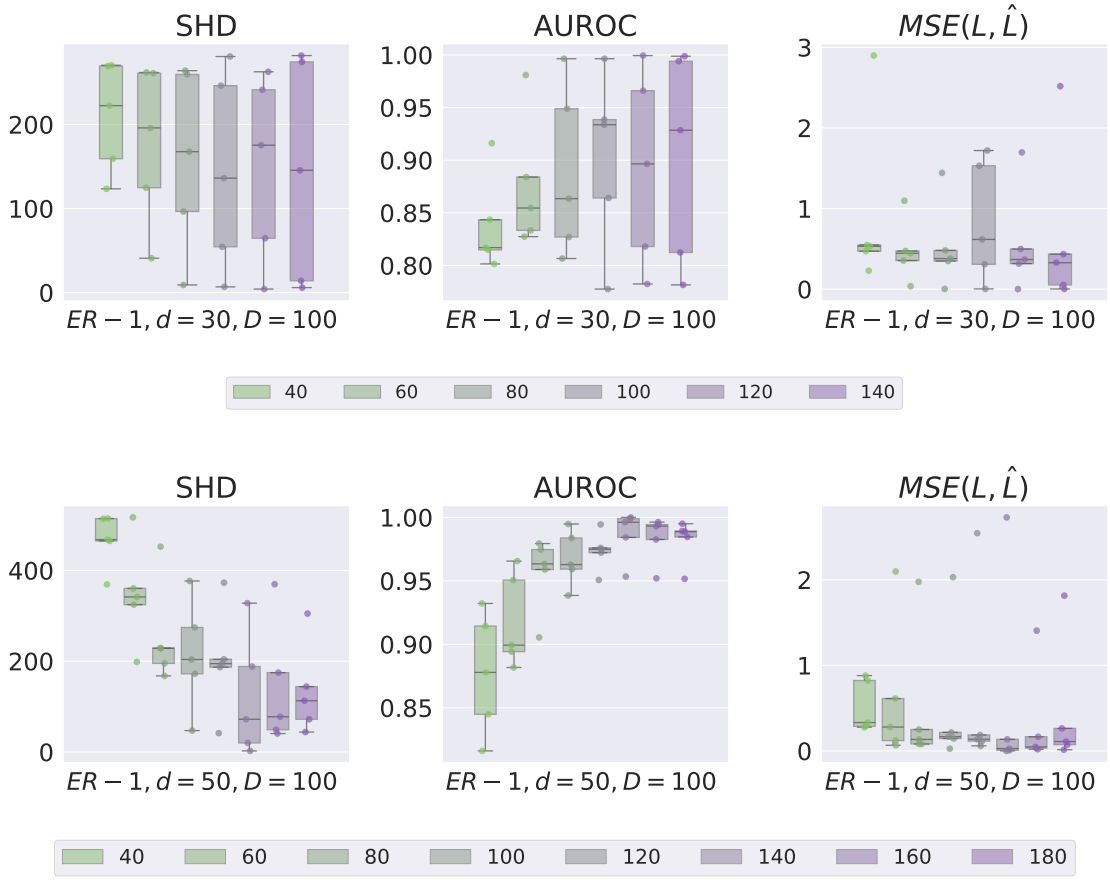

Figure 6.9: Effect of number of intervention sets on latent SCM recovery for a non-linear generation function, $d = 30, 50$ nodes. **SHD** $\downarrow$, **AUROC** $\uparrow$, **MSE**$(L, \hat{L}) \downarrow$

6.2.3 Ablation on single and multi node intervention targets

In previous experiments, we primarily consider multi-target interventions. This is done by first randomly choosing an integer k in the interval $[1, d]$. k random indices are chosen in $[0, d-1]$ and these nodes are then intervened. Though k could potentially assume the value of 1 (single-target interventions), it is usually greater, especially for larger graphs.

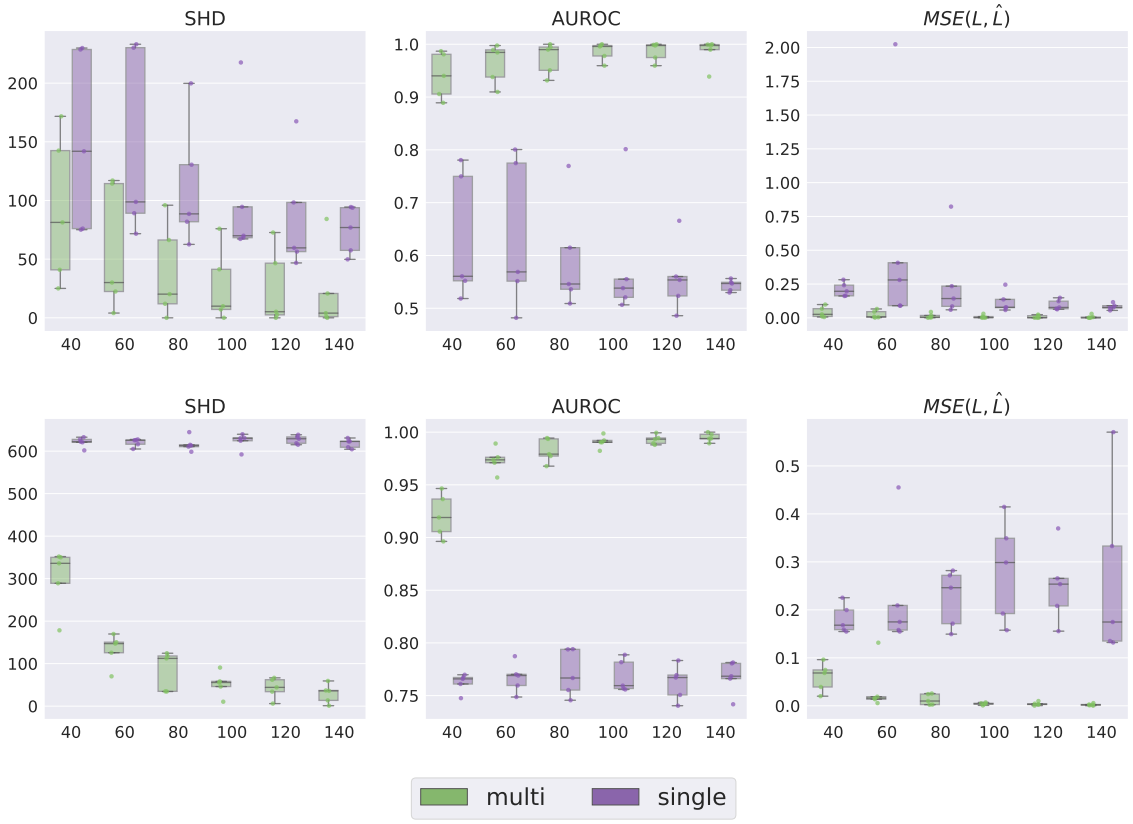

Figure 6.10: Effect of single and multi target interventional data on the latent SCM recovery, for a linear generation function, $d = 30, 50$ nodes. The X-axis refers to the number of intervention sets. **SHD** $\downarrow$, **AUROC** $\uparrow$, **MSE**$(L, \hat{L})$ $\downarrow$

In this subsection, we aim to study the effect of learning latent SCMs in the setting of single-target interventions (the setup of Brehmer et al. 2022), and how it compares to the setting of multi-target interventions.

Figure 6.11 highlights the effect of (single and multi node) intervention targets on

the latent SCM recovered by BIOLS, assuming a linear projection between latents and observed variables.

6.2.4 Ablation on range of intervention values

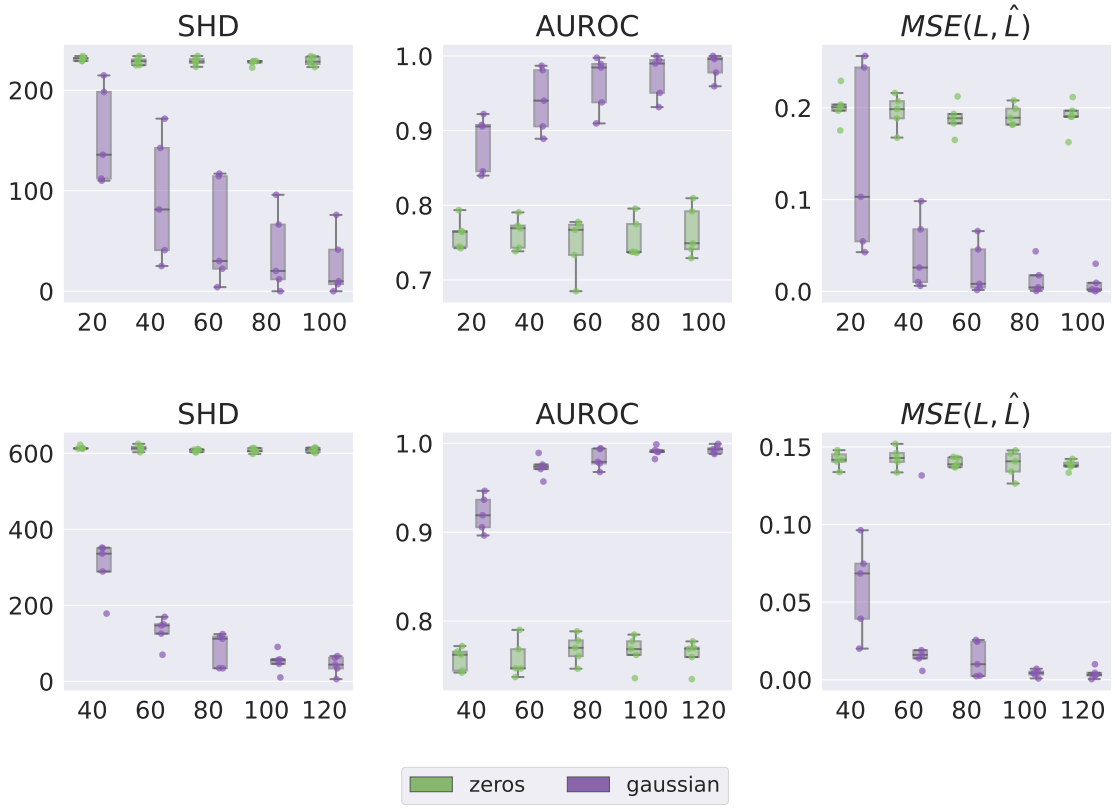

Figure 6.11: Effect of zero and gaussian intervention values on latent SCM recovery, assuming a linear generation function, $d = 30, 50$ nodes. The X-axis refers to the number of intervention sets. **SHD** $\downarrow$, **AUROC** $\uparrow$, **MSE**$(L, \hat{L}) \downarrow$

Interventional data is useful to learn more about the structure of the causal model. However, the range of values an intervened node is set to can also affect the performance of a causal discovery algorithm. Generally, one should expect the SCM recovery to be equal or better in the case where the range of intervention value is larger. To see this, consider a simple SCM: $A \to B$. Intervening on A with a large range of values gives more information about $p(B \mid A)$.

Motivated by this example, we perform an ablation study to see if and how much the range of intervention values affect the performance of BIOLS. Figure 6.11 illustrates for a linear projection function, the performance of BIOLS for various number of intervention set where the intervention value is **(i)** deterministically set to 0 or **(ii)** sampled from a Normal distribution containing a larger range of values. Figure 6.12 illustrates a similar set of results but in the case of a nonlinear projection function. Similar to previous experiments, 100 interventional samples are collected for each intervention set.

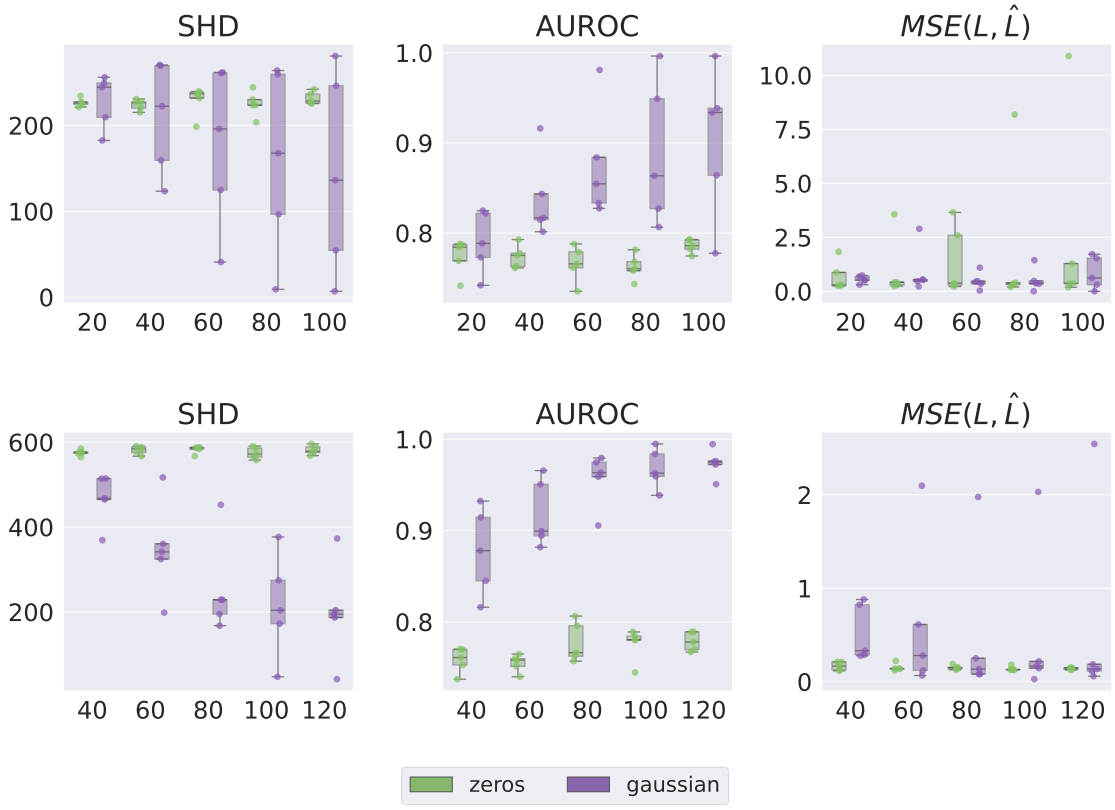

Figure 6.12: Effect of zero and gaussian intervention values on latent SCM recovery, for an nonlinear generation function modeled by a 3 layer MLP, $d = 30, 50$ nodes. The X-axis refers to the number of intervention sets. **SHD** $\downarrow$, **AUROC** $\uparrow$, **MSE**$(L, \hat{L})$ $\downarrow$

We note that moving from a Gaussian intervention to a zero-intervention setting strongly influences the performance of BIOLS. A *range* of intervention values helps latent SCM recovery, even if the variance is small.

6.2.5 Scaling the number of nodes

Thus far we have seen the performance of BIOLS in different settings such as learning from varying amounts of intervention data 6.2.2, learning from single-target and multi-target intervention data 6.2.3, and learning from zero valued (deterministic) interventions as well as from Gaussian (stochastic) interventions 6.2.4.

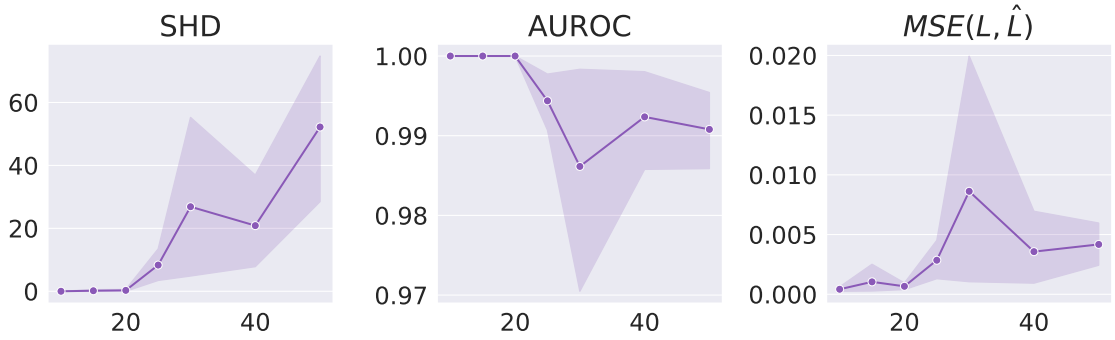

Figure 6.13: Scaling BIOLS across number of nodes for a linear data generation function, trained on multi-target interventions with Gaussian intervention values.

In this section, we focus on how BIOLS scales with the number of nodes in the SCM. Figure 6.13 plots the performance of BIOLS in a multi-target, Gaussian intervention setting. Figure 6.14 plots the performance of BIOLS in a single-target, Gaussian intervention setting. Figure 6.15 plots the performance of BIOLS in a multi-target, zero intervention setting. For all these experiments, we use 100 interventional sets with 100 data points per intervention set.

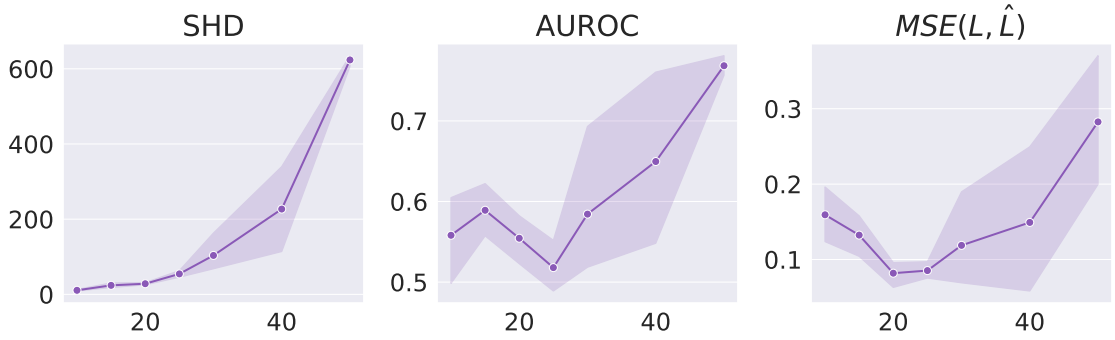

Figure 6.14: Scaling BIOLS across number of nodes for a linear data generation function, trained on single-target interventions with Gaussian intervention values.

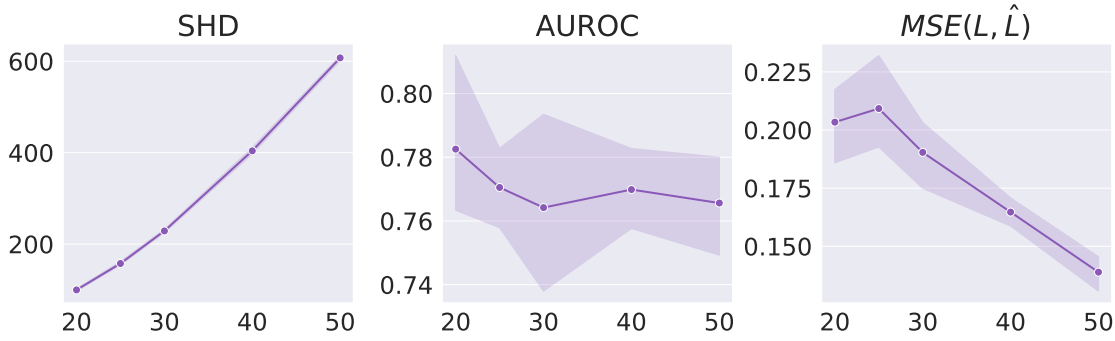

Figure 6.15: Scaling BIOLS across number of nodes for a linear data generation function, trained on multi-target interventions with intervention values fixed to 0.

In figure 6.16, we summarize results for a similar experiment, where the projection function is nonlinear. The only difference in this experiment is that we use 400 interventional sets with 100 samples per set. We note that **BIOLS successfully scales to atleast upto** 50 **nodes**.

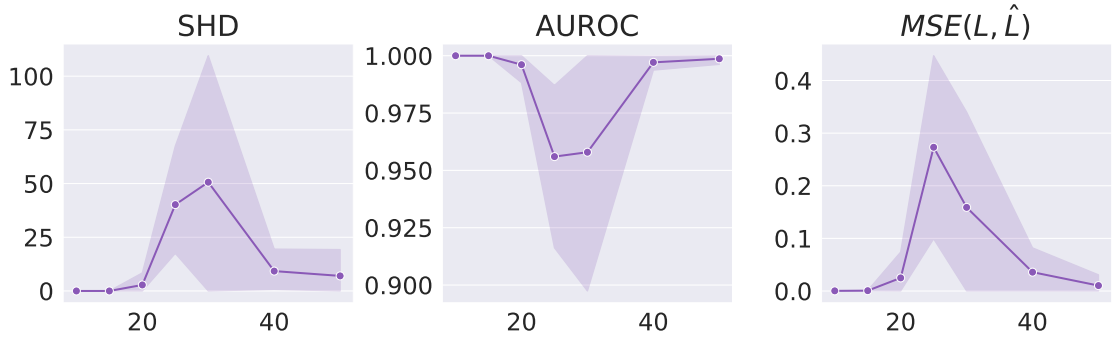

Figure 6.16: Scaling BIOLS across number of nodes for a linear data generation function, trained on multi-target interventions with Gaussian intervention values.

6.2.6 Implementation details

$SO(n)$ **projection**: The construction of the projection matrix follows the methodology outlined in (Brehmer et al. 2022). However, for the sake of completeness, we provide a detailed description of the steps involved in obtaining this projection matrix. First, coefficients c_{ij} are drawn from a Normal distribution, $c_{ij} \sim \mathcal{N}(0, 0.05^2)$, for every entry where $j < i$ in the lower triangle of a $\mathbb{R}^{d \times d}$ matrix. Subsequently, to ensure skew-symmetry, the upper triangular entries are populated with values such that $c_{ji} = -c_{ij}$. Finally, the matrix exponentiation process is applied to yield the desired projection matrix. This method ensures the matrix conforms to the special orthogonal group, $SO(n)$.

Nonlinear projection: A random 3-layer neural network with ReLU activations is initialized to execute the projection from d to D dimensions. While our reported experimental results focus on ReLU activations, it's important to note that BIOLS is versatile and supports nonlinear projections involving alternative activation functions, such as leaky ReLU, GeLU, among others. The network sizes for the 3-layer MLP are specified in Table 6.1. In all our experiments, D is set to 100. For all our experiments, we use the AdaBelief (Zhuang et al. 2020) optimizer with $\epsilon = 10^{-8}$ and a learning

rate of 0.0008. Our experiments are fairly robust with respect to hyperparameters and we did not perform hyperparameter tuning for any of our experiments. Table 6.2 summarizes the network details for the generative model $p_\psi(\mathcal{X} \mid \mathcal{Z})$.

Table 6.1: Network architecture for the nonlinear projection

Layer type	**Layer output**	**Activation**
Linear	D	GeLU
Linear	D	GeLU
Linear	D	

Table 6.2: Network architecture for the decoder $p_\psi(\mathcal{X} \mid \mathcal{Z})$

Layer type	**Layer output**	**Activation**
Linear	16	GeLU
Linear	64	GeLU
Linear	64	GeLU
Linear	64	GeLU
Linear	D	

Ancestral sampling from $q_\phi(L, \Sigma)$: In section 5.3, we had mentioned that the posterior $q_\phi(L, \Sigma)$ is a K-variate Normal distribution with a diagonal covariance, where $K = \frac{d(d+1)}{2}$. This corresponds to a distribution over the $\frac{d(d-1)}{2}$ edges over the DAG (i.e, denoted by L, the lower triangular elements) and d additional elements corresponding to exogenous noise variables $\epsilon = [\epsilon_1, \ldots, \epsilon_d]^T$ of the latent SCM (denoted by Σ). Suppose that the parameters of the Gaussian $q_\phi(L, \Sigma)$ is given by mean $\boldsymbol{\mu}_q$ and precision $\boldsymbol{P}_q$:

$$\boldsymbol{\mu}_q = (\boldsymbol{\mu}_L, \boldsymbol{\mu}_{\boldsymbol{\Sigma}}) \qquad \boldsymbol{\mu}_L \in \mathbb{R}^{(K-d)} \quad \boldsymbol{\mu}_\Sigma \in \mathbb{R}^d \tag{6.1}$$

$$\boldsymbol{P}_q^{-1} = \begin{bmatrix} \boldsymbol{P}_L^{-1} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{P}_\Sigma^{-1} \end{bmatrix} \qquad \boldsymbol{P}_L^{-1} \in \mathbb{R}^{(K-d)\times(K-d)} \quad \boldsymbol{P}_\Sigma^{-1} \in \mathbb{R}^{d\times d} \tag{6.2}$$

Consider $S \sim q_\phi(L, \Sigma)$. The first $K - d$ elements represent the weighted adjacency matrix $\widehat{W}$; for this, populate these $K - d$ elements in the lower triangle of a zero

matrix. The last d elements, $S_{(K-d):K}$, are samples of the predicted exogenous noise variables $\hat{\epsilon}$. Now, the SCM is defined by $\widehat{W}$ and $\hat{\epsilon}$, and ancestral sampling is expressed as $z_i := \widehat{W}_{*i}^T \mathbf{z} + \hat{\epsilon}_i$, where the ordering of assignment indexed by i is according to the topological ordering induced by $\widehat{W}$. z_i is the i^{th} element of $\mathbf{z}$, and $\mathbf{z}$ is initialized to zeros. $\widehat{W}_{*i}$ denotes the i^{th} column vector of $\widehat{W}$.

6.2.7 Additional Visualizations

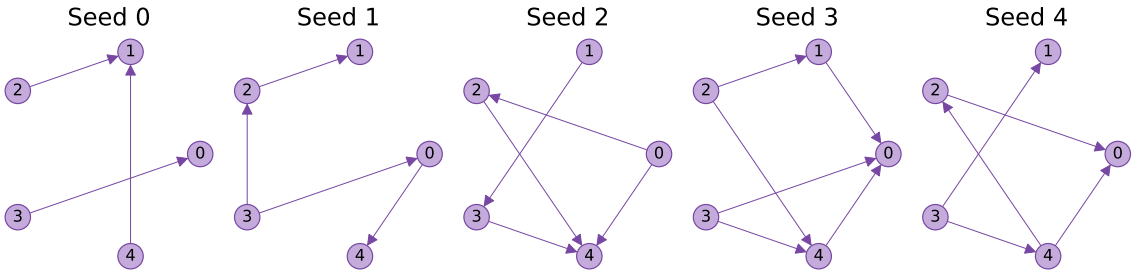

Figure 6.17: Ground truth causal structures for the experiment on the chemistry dataset.

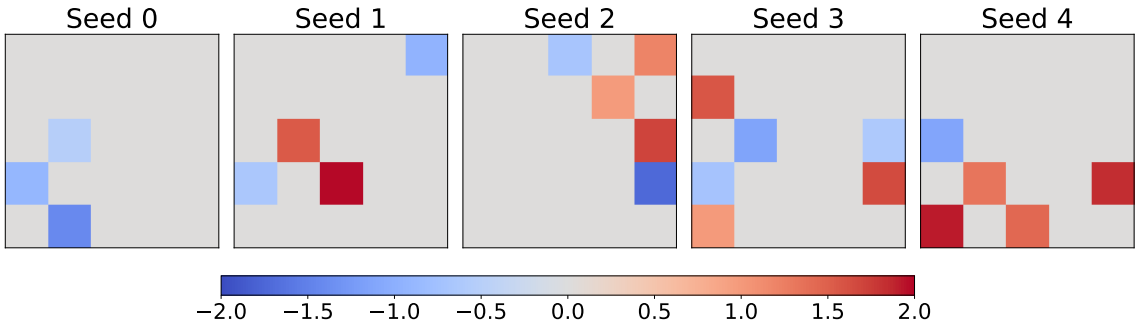

Figure 6.18: Ground truth weighted adjacency matrices for the experiment on the chemistry dataset.

6.2.8 Runtimes

In this subsection, we note down the runtimes for a subset of the previously presented experiments. We explore the program runtime along two axes: scaling with respect to nodes, and with respect to number of data points.

	$n = 4000$	$n = 6000$	$n = 8000$	$n = 10000$	$n = 12000$
$d = 15$	12	15	17	20	23
$d = 20$	20	24	26	29	35
$d = 25$	28	32	36	40	45
$d = 30$	40	47	50	55	60
$d = 40$	72	76	90	100	105
$d = 50$	105	116	127	142	166

Table 6.3: Program runtimes: Scaling BIOLS across number of nodes and data points, with $D = 100$. All runs are reported on 10000 epochs of BIOLS across 5 seeds. All reported runtimes are in minutes.

Table 6.3 presents the program runtime for scaling BIOLS, where n refers to pairs of $(\text{observational}, \text{interventional})$ data and d refers to the number of nodes in the latent SCM. All runs are reported on 10000 epochs of BIOLS across 5 seeds, with the observed dimensions D set to 100 and a linear projection function. The runtimes remain similar for nonlinear projection since the neural network sizes remain the same, and it is only the data generation procedure that is different.

7

Conclusion

In this thesis, we studied the problem of causal discovery in the latent space which has close connections to causal representation learning. A number of assumptions have made this possible, such as known intervention targets and values, assuming a fixed node ordering, and that the class of latent SCMs belongs to linear Gaussian models. Unlike other works, we assume the intervention values are sampled from some distribution (such as Normal or Uniform) instead of always using 0-valued interventions.

We presented a tractable approximate inference technique to perform Bayesian latent causal discovery that jointly infers the causal variables, structure and parameters of linear Gaussian latent SCMs under random, known interventions from low-level data (such as high-dimensional vectors or pixels).

Though Brehmer et al. 2022 addresses a closely related problem, it is limited to the realm of single-target, zero valued interventions, that require paired counterfactual data. In contrast, BIOLS also supports multi-target, non-zero valued interventions that neither require the data to be paired or counterfactual.

The advantage of the learned causal model, BIOLS, is also shown. BIOLS exhibits generalization by sampling images from unseen interventional distributions. Furthermore, the Bayesian formulation allows for uncertainty estimation. This can be used to obtain estimates of mutual information gain, as a result of altering the current belief of a causal model. This makes our formulation particularly well-suited for extensions

to active causal discovery.

However, answering questions such as *how to efficiently perform intervention inference from low-level data* remains challenging. Brehmer et al. 2022 provides some direction towards answering this question, but can infer only single target interventions (search space d). Whereas, the multi-target intervention inference problem has a much larger search space (2^d).

Finally, ILCM takes very long to train, requires more data points, and gets hard to train beyond $8-10$ nodes. In contrast, BIOLS scales better and faster (till upto atleast $20-50$ nodes), as long there is enough interventional data to recover the latent SCM. A reinforcement learning setup where the agent can actively experiment with the environment to understand the world could be yet another area to explore, in order to better learn the latent SCM and intervention targets. Extensions of the proposed method to learn nonlinear, non-Gaussian latent SCMs from unknown interventions would open doors to more general algorithms that can learn causal representations.

Bibliography

Agrawal, Raj et al. (2019). "ABCD-Strategy: Budgeted Experimental Design for Targeted Causal Structure Discovery". In: *International Conference on Artificial Intelligence and Statistics*.

Ahuja, Kartik, Jason Hartford, and Yoshua Bengio (2022). "Weakly Supervised Representation Learning with Sparse Perturbations". In: *arXiv preprint arXiv:2206.01101*.

Ahuja, Kartik, Yixin Wang, et al. (2022). "Interventional Causal Representation Learning". In: *ArXiv* abs/2209.11924.

Akrout, Mohamed et al. (2019). "Deep Learning without Weight Transport". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc.

Anandkumar, Animashree et al. (2012). "Learning topic models and latent Bayesian networks under expansion constraints". In: *arXiv preprint arXiv:1209.5350*.

Annadani, Yashas et al. (2021). "Variational causal networks: Approximate bayesian inference over causal structures". In: *arXiv preprint arXiv:2106.07635*.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473.

Barron, A., J. Rissanen, and Bin Yu (1998). "The minimum description length principle in coding and modeling". In: *IEEE Transactions on Information Theory* 44.6, pp. 2743–2760.

Bengio, Emmanuel et al. (2021). "Flow network based generative models for non-iterative diverse candidate generation". In: *Advances in Neural Information Processing Systems* 34, pp. 27381–27394.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent (2013). "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8, pp. 1798–1828.

Bengio, Yoshua, Tristan Deleu, et al. (2019). "A meta-transfer objective for learning to disentangle causal mechanisms". In: *arXiv preprint arXiv:1901.10912*.

Bengio, Yoshua, Nicholas Léonard, and Aaron C. Courville (2013). "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation". In: *ArXiv* abs/1308.3432.

Blei, David M, Alp Kucukelbir, and Jon D McAuliffe (2017). "Variational inference: A review for statisticians". In: *Journal of the American statistical Association* 112.518, pp. 859–877.

Bradbury, James et al. (2018). "JAX: composable transformations of Python+ NumPy programs". In: *Version 0.2* 5, pp. 14–24.

Brehmer, Johann et al. (2022). "Weakly supervised causal representation learning". In: *Advances in Neural Information Processing Systems*. Vol. 35.

Brouillard, Philippe et al. (2020). "Differentiable causal discovery from interventional data". In: *Advances in Neural Information Processing Systems* 33, pp. 21865–21877.

Carvalho, Carlos M., Nicholas G. Polson, and James G. Scott (16–18 Apr 2009). "Handling Sparsity via the Horseshoe". In: *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*. Ed. by David van Dyk and Max Welling. Vol. 5. Proceedings of Machine Learning Research. Hilton Clearwater Beach Resort, Clearwater Beach, Florida USA: PMLR, pp. 73–80.

Cheng, Jie et al. (2002). "Learning Bayesian Networks From Data: An Information-Theory Based Approach". In: *Artificial Intelligence* 137.1-2, pp. 43–90.

Chickering, David Maxwell (2002). "Optimal structure identification with greedy search". In: *Journal of machine learning research* 3.Nov, pp. 507–554.

Cho, Kyunghyun et al. (2014). "On the properties of neural machine translation: Encoder-decoder approaches". In: *arXiv preprint arXiv:1409.1259*.

Cundy, Chris, Aditya Grover, and Stefano Ermon (2021). "BCD Nets: Scalable Variational Approaches for Bayesian Causal Discovery". In: *Neural Information Processing Systems*.

Deleu, Tristan, Ant'onio G'ois, et al. (2022). "Bayesian Structure Learning with Generative Flow Networks". In: *Conference on Uncertainty in Artificial Intelligence*.

Deleu, Tristan, Mizu Nishikawa-Toomey, et al. (2023). "Joint Bayesian Inference of Graphical Structure and Parameters with a Single Generative Flow Network". In: *Advances in Neural Information Processing Systems*.

Diallo, Aïssatou, Markus Zopf, and Johannes Fürnkranz (2020). "Permutation Learning via Lehmer Codes". In: *European Conference on Artificial Intelligence*.

Dreyfus, Stuart (1962). "The numerical solution of variational problems". In: *Journal of Mathematical Analysis and Applications* 5.1, pp. 30–45.

Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *Journal of Machine Learning Research* 12.61, pp. 2121–2159.

Elidan, Gal et al. (2000). "Discovering Hidden Variables: A Structure-Based Approach". In: *Advances in Neural Information Processing Systems*. Ed. by T. Leen, T. Dietterich, and V. Tresp. Vol. 13. MIT Press.

Elsayed, Gamaleldin F. et al. (2022). "SAVi++: Towards end-to-end object-centric learning from real-world videos". In: *Advances in Neural Information Processing Systems (NeurIPS)*.

Erdos, Paul, Alfréd Rényi, et al. (1960). "On the evolution of random graphs". In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1, pp. 17–60.

Fay, Marta M et al. (2023). "RxRx3: Phenomics Map of Biology". In: *bioRxiv*, pp. 2023–02.

Friedman, Nir (1998). "The Bayesian Structural EM Algorithm". In: *Conference on Uncertainty in Artificial Intelligence*.

Friedman, Nir and Daphne Koller (2013). "Being Bayesian about network structure". In: *arXiv preprint arXiv:1301.3856*.

Fukushima, Kunihiko (1980). "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position". In: *Biological cybernetics* 36.4, pp. 193–202.

Geiger, Dan and David Heckerman (1994). "Learning Gaussian Networks". In: *UAI*.

Ghoshal, Asish and Jean Honorio (2018). "Learning linear structural equation models in polynomial time and sample complexity". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1466–1475.

Goodfellow, Ian J. et al. (2014). "Generative Adversarial Nets". In: *NIPS*.

Goyal, Anirudh and Yoshua Bengio (2022). "Inductive biases for deep learning of higher-level cognition". In: *Proceedings of the Royal Society A* 478.2266, p. 20210068.

Hägele, Alexander et al. (2022). "BaCaDI: Bayesian Causal Discovery with Unknown Interventions". In: *arXiv preprint arXiv:2206.01665*.

Hausman, Daniel M. and James Woodward (1999). "Independence, Invariance and the Causal Markov Condition". In: *The British Journal for the Philosophy of Science* 50.4, pp. 521–583.

He, Jiawei et al. (2019). "Variational Autoencoders with Jointly Optimized Latent Dependency Structure". In: *International Conference on Learning Representations*.

Hebb, Donald O. (1949). *The organization of behavior: A neuropsychological theory*. Wiley.

Heckerman, David, Dan Geiger, and David M Chickering (1995). "Learning Bayesian networks: The combination of knowledge and statistical data". In: *Machine learning* 20, pp. 197–243.

Heckerman, David, Christopher Meek, and Gregory Cooper (1997). *A Bayesian approach to causal discovery*. Tech. rep. Technical report msr-tr-97-05, Microsoft Research.

— (2006). "A Bayesian approach to causal discovery". In: *Innovations in Machine Learning: Theory and Applications*, pp. 1–28.

Helmbold, David and Manfred Warmuth (Sept. 2009). "Learning Permutations with Exponential Weights". In: vol. 10, pp. 1705–1736.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780.

Hoyer, Patrik et al. (2008). "Nonlinear causal discovery with additive noise models". In: *Advances in neural information processing systems* 21.

Jang, Eric, Shixiang Shane Gu, and Ben Poole (2016). "Categorical Reparameterization with Gumbel-Softmax". In: *ArXiv* abs/1611.01144.

Kahneman, Daniel (2011). *Thinking, fast and slow*. Macmillan.

Kass, Robert E. and Adrian E. Raftery (1995). "Bayes Factors". In: *Journal of the American Statistical Association* 90.430, pp. 773–795.

Ke, Nan Rosemary, Olexa Bilaniuk, et al. (2019). "Learning Neural Causal Models from Unknown Interventions". In: *arXiv preprint arXiv:1910.01075*.

Ke, Nan Rosemary, Silvia Chiappa, et al. (2022). "Learning to Induce Causal Structure". In: *arXiv preprint arXiv:2204.04875*.

Ke, Nan Rosemary, Aniket Didolkar, et al. (2021). "Systematic evaluation of causal discovery in visual model based reinforcement learning". In: *arXiv preprint arXiv:2107.00848*.

Ke, Nan Rosemary, Sara-Jane Dunn, et al. (2023). "DiscoGen: Learning to Discover Gene Regulatory Networks". In: *arXiv preprint arXiv:2304.05823*.

Kelley, Henry J. (1960). "Gradient Theory of Optimal Flight Paths". In: *ARS Journal* 30, pp. 947–954.

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Kingma, Diederik P. and Jimmy Ba (2014). "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980.

Kingma, Durk P, Tim Salimans, et al. (2016). "Improved variational inference with inverse autoregressive flow". In: *Advances in neural information processing systems* 29.

Kivva, Bohdan et al. (2021). "Learning latent causal graphs via mixture oracles". In: *Advances in Neural Information Processing Systems* 34, pp. 18087–18101.

Kocaoglu, Murat et al. (2018). "CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training". In: *International Conference on Learning Representations*.

Koivisto, Mikko and Kismat Sood (2004). "Exact Bayesian structure discovery in Bayesian networks". In: *The Journal of Machine Learning Research* 5, pp. 549–573.

Kuhn, Harold W (1955). "The Hungarian method for the assignment problem". In: *Naval research logistics quarterly* 2.1-2, pp. 83–97.

Kuipers, Jack, Polina Suter, and Giusi Moffa (2022). "Efficient sampling and structure learning of Bayesian networks". In: *Journal of Computational and Graphical Statistics*, pp. 1–12.

Lachapelle, Sébastien et al. (2019). "Gradient-based neural dag learning". In: *arXiv preprint arXiv:1906.02226*.

LeCun, Y. et al. (1989). "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4, pp. 541–551.

Lee, Dong-Hyun et al. (2014). "Difference Target Propagation". In: *ECML/PKDD*.

Lehmann, E. L. and Joseph P. Romano (2005). *Testing statistical hypotheses*. Springer.

Li, Yuke et al. (2022). *Intervention-based Recurrent Causal Model for Non-stationary Video Causal Discovery*.

Lippe, Phillip, Taco Cohen, and Efstratios Gavves (2022). "Efficient Neural Causal Discovery without Acyclicity Constraints". In: *International Conference on Learning Representations*.

Lippe, Phillip, Sara Magliacane, et al. (2022). "CITRIS: Causal Identifiability from Temporal Intervened Sequences". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 13557–13603.

Locatello, Francesco et al. (2020). "Object-Centric Learning with Slot Attention". In: *ArXiv* abs/2006.15055.

Lopez-Paz, David, Robert Nishihara, et al. (2017). "Discovering causal signals in images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6979–6987.

Lopez-Paz, David and Maxime Oquab (2016). "Revisiting Classifier Two-Sample Tests". In: *arXiv: Machine Learning*.

Lorch, Lars et al. (2021). "DiBS: Differentiable Bayesian Structure Learning". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. Vol. 34. Curran Associates, Inc., pp. 24111–24123.

Markham, Alex and Moritz Grosse-Wentrup (Mar. 2020). "Measurement Dependence Inducing Latent Causal Models". In: *Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI)*. Ed. by Jonas Peters and David Sontag. Vol. 124. Proceedings of Machine Learning Research. PMLR, pp. 590–599.

Matthews, Robert (2000). "Storks Deliver Babies (p= 0.008)". In: *Teaching Statistics*.

McCulloch, Warren and Walter Pitts (1943). "A Logical Calculus of Ideas Immanent in Nervous Activity". In: *Bulletin of Mathematical Biophysics*.

Minsky, Marvin and Seymour Papert (1969). *Perceptrons: An Introduction to Computational Geometry*. MIT Press.

Moraffah, Raha et al. (2020). "Causal adversarial network for learning conditional and interventional distributions". In: *arXiv preprint arXiv:2008.11376*.

Nair, Vinod and Geoffrey E. Hinton (2010). "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *International Conference on Machine Learning*.

Ng, Ignavier, AmirEmad Ghassami, and Kun Zhang (2020). "On the role of sparsity and dag constraints for learning linear dags". In: *Advances in Neural Information Processing Systems* 33, pp. 17943–17954.

Ng, Ignavier, Shengyu Zhu, et al. (2022). "Masked gradient-based causal structure learning". In: *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*. SIAM, pp. 424–432.

Nishikawa-Toomey, Mizu et al. (2023). "Bayesian learning of Causal Structure and Mechanisms with GFlowNets and Variational Bayes". In: *AAAI Workshop Graphs and More Complex Structures for Learning and Reasoning*.

Pamfil, Roxana et al. (2020). "Dynotears: Structure learning from time-series data". In: *International Conference on Artificial Intelligence and Statistics*. PMLR, pp. 1595–1605.

Pascanu, Razvan, Tomas Mikolov, and Yoshua Bengio (2013). "On the difficulty of training recurrent neural networks". In: *International conference on machine learning*. Pmlr, pp. 1310–1318.

Pearl, Judea (2009a). *Causality*. 2nd ed. Cambridge University Press.

— (2009b). *Causality: Models, Reasoning and Inference*. Cambridge University Press.

Peters, J., Joris M. Mooij, et al. (2013). "Causal discovery with continuous additive noise models". In: *J. Mach. Learn. Res.* 15, pp. 2009–2053.

Peters, Jonas and Peter Bühlmann (2012). "Identifiability of Gaussian structural equation models with equal error variances". In: *Biometrika*.

— (2014). "Identifiability of Gaussian structural equation models with equal error variances". In: *Biometrika* 101.1, pp. 219–228.

Rezende, Danilo Jimenez, Shakir Mohamed, and Daan Wierstra (22–24 Jun 2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Mod-

els". In: *Proceedings of the 31st International Conference on Machine Learning*. Ed. by Eric P. Xing and Tony Jebara. Vol. 32. Proceedings of Machine Learning Research 2. Bejing, China: PMLR, pp. 1278–1286.

Robbins, Herbert E. (1951). "A Stochastic Approximation Method". In: *Annals of Mathematical Statistics* 22, pp. 400–407.

Rosenblatt, F. (1958). "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review*.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986a). "Learning Internal Representations by Error Propagation". In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA: MIT Press, pp. 318–362.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams (1986b). "Learning Representations by Back-propagating Errors". In: *Nature* 323.6088, pp. 533–536.

Sak, Hasim, Andrew W. Senior, and Françoise Beaufays (2014). "Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition". In: *ArXiv* abs/1402.1128.

Scherrer, Nino, Olexa Bilaniuk, et al. (2021). "Learning neural causal models with active interventions". In: *arXiv preprint arXiv:2109.02429*.

Scherrer, Nino, Anirudh Goyal, et al. (2022). "On the generalization and adaption performance of causal models". In: *arXiv preprint arXiv:2206.04620*.

Schölkopf, Bernhard, Dominik Janzing, et al. (2012). "On causal and anticausal learning". In: *arXiv preprint arXiv:1206.6471*.

Schölkopf, Bernhard, Francesco Locatello, et al. (2021a). "Toward causal representation learning". In: *Proceedings of the IEEE*.

— (2021b). "Toward causal representation learning". In: *Proceedings of the IEEE* 109.5, pp. 612–634.

Shen, Xinwei et al. (2022). "Weakly Supervised Disentangled Generative Causal Representation Learning". In: *Journal of Machine Learning Research* 23.241, pp. 1–55.

Shimizu, Shohei et al. (2011). "DirectLiNGAM: A direct method for learning a linear non-Gaussian structural equation model". In: *The Journal of Machine Learning Research* 12, pp. 1225–1248.

Silva, Ricardo et al. (2006). "Learning the Structure of Linear Latent Variable Models". In: *Journal of Machine Learning Research* 7.8, pp. 191–246.

Sinkhorn, Richard (1964). "A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices". In: *Annals of Mathematical Statistics* 35, pp. 876–879.

Sønderby, Casper Kaae et al. (2016). "Ladder Variational Autoencoders". In: *NIPS*.

Spirtes, P., C. Glymour, and R. Scheines (2000). *Causation, Prediction, and Search*. MIT press.

Spirtes, Peter, Clark N Glymour, et al. (2000). *Causation, prediction, and search*. MIT press.

Su, Jiawei, Danilo Vasconcellos Vargas, and Kouichi Sakurai (2019). "One Pixel Attack for Fooling Deep Neural Networks". In: *IEEE Transactions on Evolutionary Computation*.

Tigas, Panagiotis et al. (2022). "Interventions, where and how? experimental design for causal models at scale". In: *arXiv preprint arXiv:2203.02016*.

Toth, Christian et al. (2022). "Active Bayesian Causal Inference". In: *arXiv preprint arXiv:2206.02063*.

Turing, A. M. (1950). "Computing Machinery and Intelligence". In: *Mind*.

Vaswani, Ashish et al. (2017). "Attention is all you need". In: *Advances in neural information processing systems* 30.

Viinikka, Jussi et al. (2020). "Towards scalable bayesian learning of causal dags". In: *Advances in Neural Information Processing Systems* 33, pp. 6584–6594.

Wang, Benjie, Matthew R Wicker, and Marta Kwiatkowska (2022). "Tractable Uncertainty for Structure Learning". In: *International Conference on Machine Learning*. PMLR, pp. 23131–23150.

Wang, Yixin and Michael I Jordan (2021). "Desiderata for representation learning: A causal perspective". In: *arXiv preprint arXiv:2109.03795*.

Wei, Dennis, Tian Gao, and Yue Yu (2020). "DAGs with No Fears: A closer look at continuous optimization for learning Bayesian networks". In: *Advances in Neural Information Processing Systems* 33, pp. 3895–3906.

Wright, Sewall (1918). "On the Nature of Size Factors". In: *Genetics*.

— (1934). "The method of path coefficients". In: *The annals of mathematical statistics*.

Xie, Feng, Ruichu Cai, et al. (2020). "Generalized independent noise condition for estimating latent variable causal graphs". In: *Advances in Neural Information Processing Systems* 33, pp. 14891–14902.

Xie, Feng, Biwei Huang, et al. (17–23 Jul 2022). "Identification of Linear Non-Gaussian Latent Hierarchical Structure". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri et al. Vol. 162. Proceedings of Machine Learning Research. PMLR, pp. 24370–24387.

Yang, Mengyue et al. (2021). "CausalVAE: Disentangled representation learning via neural structural causal models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9593–9602.

Yu, Yue, Jie Chen, et al. (2019). "DAG-GNN: DAG Structure Learning with Graph Neural Networks". In: *International Conference on Machine Learning*.

Yu, Yue, Tian Gao, et al. (2021). "Dags with no curl: An efficient dag structure learning approach". In: *International Conference on Machine Learning*. PMLR, pp. 12156–12166.

Zhang, Zhen et al. (2022). "Truncated Matrix Power Iteration for Differentiable DAG Learning". In: *Advances in Neural Information Processing Systems*.

Zhao, Shengjia, Jiaming Song, and Stefano Ermon (June 2017). "Learning Hierarchical Features from Deep Generative Models". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 4091–4099.

Zheng, Xun et al. (2018). "Dags with no tears: Continuous optimization for structure learning". In: *Advances in Neural Information Processing Systems* 31.

Zhuang, Juntang et al. (2020). "Adabelief optimizer: Adapting stepsizes by the belief in observed gradients". In: *Advances in neural information processing systems* 33, pp. 18795–18806.