entropy

MDPI

Article

Diffusion-Based Causal Representation Learning

Amir Mohammad Karimi Mamaghan [1,*], Andrea Dittadi [2,3,4], Stefan Bauer [2,4], Karl Henrik Johansson [1,5] and Francesco Quinzan [6]

[1] Division of Decision and Control Systems (DCS), KTH Royal Institute of Technology, 114 28 Stockholm, Sweden; kallej@kth.se
[2] Helmholtz AI, 85764 Munich, Germany; andrea.dittadi@helmholtz-munich.de (A.D.); stefan.bauer@helmholtz-munich.de (S.B.)
[3] MPI for Intelligent Systems, 72076 Tübingen, Germany
[4] School of Computation, Information and Technology, TU Munich, 80333 Munich, Germany
[5] Digital Futures, 114 28 Stockholm, Sweden
[6] Department of Computer Science, University of Oxford, Oxford OX1 2JD, UK; francesco.quinzan@cs.ox.ac.uk
* Correspondence: amkm@kth.se

Abstract: Causal reasoning can be considered a cornerstone of intelligent systems. Having access to an underlying causal graph comes with the promise of cause–effect estimation and the identification of efficient and safe interventions. However, learning causal representations remains a major challenge, due to the complexity of many real-world systems. Previous works on causal representation learning have mostly focused on Variational Auto-Encoders (VAEs). These methods only provide representations from a point estimate, and they are less effective at handling high dimensions. To overcome these problems, we propose a Diffusion-based Causal Representation Learning (DCRL) framework which uses diffusion-based representations for causal discovery in the latent space. DCRL provides access to both single-dimensional and infinite-dimensional latent codes, which encode different levels of information. In a first proof of principle, we investigate the use of DCRL for causal representation learning in a weakly supervised setting. We further demonstrate experimentally that this approach performs comparably well in identifying the latent causal structure and causal variables.

Keywords: diffusion models; diffusion-based representations; causal representation learning; weak supervision

check for updates

Citation: Karimi Mamaghan, A.M.; Dittadi, A.; Bauer, S.; Johansson, K.H.; Quinzan, F. Diffusion-Based Causal Representation Learning. *Entropy* **2024**, *26*, 556. https://doi.org/10.3390/e26070556

Academic Editor: Jakub Tomczak

Received: 30 May 2024
Revised: 24 June 2024
Accepted: 25 June 2024
Published: 28 June 2024

1. Introduction

Causal representation learning consists of uncovering a system's latent causal factors and their relationships, from observed low-level data. It finds applicability in domains such as autonomous driving [1], robotics [2], healthcare [3], climate studies [4], epidemiology [5,6], and finance [7]. Furthermore, recent advancements in Large Language Models (LLMs) underscore the growing importance of studying causal representation learning in this domain [8–10]. In these tasks, the underlying causal variables are often unknown, and we only have access to low-level representations.

Causal representation learning is a challenging problem. In fact, identifying latent causal factors is generally impossible from observational data only. There has been an ongoing effort to study sets of assumptions that ensure the identifiability of causal variables and their relationships [1,11–17]. These approaches consider the availability of additional information, or they use assumptions on the underlying causal structure of the DGP. However, many of these assumptions, such as Causal Faithfulness [18] cannot be verified. However, it is possible to identify latent causal factors from observational *and* interventional data. Brehmer et al. [14] considers a weak form of supervision, in which we have access to a data pair, corresponding to the state of the system before and after a random unknown intervention. Brehmer et al. [14] proves that, in this weakly supervised setting, the structure and the causal variables are identifiable up to a relabeling and elementwise reparameterization.

There has been a growing interest in leveraging generative models to learn causal representations with specific properties. For example, disentangled and object-centric representations have been shown to be helpful for complex downstream tasks and generalization [19–24]. Variational Autoencoders (VAEs) [25] are among the most widely studied generative models, and they have been successfully used for disentanglement and causal representation learning [14,26]. However, the problem of learning causal representations has not yet been approached with more powerful generative models.

Recently, diffusion models have emerged as state-of-the-art generative models, and they have demonstrated remarkable success across several domains such as image, video, and audio synthesis [27–37], molecular generation [38–41], and representation learning [42–48]. Diffusion models draw on concepts and principles from diffusion processes to learn the data distribution [49–53]. These models exploit diffusion behavior to produce diverse, high-quality, and realistic samples. Furthermore, unlike other generative models like VAEs that encode the information in one single code, diffusion-based models have the appealing property of infinite-dimensional latent codes which contain different levels of information at different timesteps [43]. However, despite this advantage and their remarkable performance, diffusion models have not yet been employed for causal representation learning, indicating that their potential has yet to be explored in this context.

In this work, we study the connection between diffusion-based models and causal structure learning by employing representations obtained from diffusion models for the task of causal representation learning. In particular, our contributions are the following:

- We propose *DCRL*, a diffusion-based framework for causal representation learning in weakly supervised settings.
- We derive the Evidence Lower Bound (ELBO) for DCRL, in the case of both finite- and infinite-dimensional representations.
- We empirically illustrate that the noise- and diffusion-based representations contain equivalent information about the underlying causal variables and causal mechanisms, and can be used interchangeably.

The rest of the paper is organized as follows: Section 2 explains the related works. Section 3 covers the background on causality and diffusion models. The background on diffusion models and diffusion-based representations are outlined in Section 4. Section 5 outlines the addressed problem, the weakly supervised framework, and the identifiability conditions. Section 6 details the proposed DCRL framework. Experimental results are presented in Section 7. Finally, Section 8 concludes the paper and suggests potential future research directions.

2. Related Work

2.1. Diffusion-Based Representation Learning

Learning representations with diffusion models remains a relatively unexplored area. Several works have tried to train an external module (e.g., an encoder) along with the score function of the diffusion model to extract representations. Abstreiter et al. [43] and Mittal et al. [44] condition the score function of a diffusion model on a time-independent and time-dependent encoder and obtain finite and infinite-dimensional representations, respectively. Wang et al. [45] uses the same conditioning but regularizes the objective function with the mutual information between the input data and learned representations. Traub [48] performs the same conditioning but the authors use Latent Diffusion Models [54], where the inputs of the diffusion model are latent variables obtained from applying a pre-trained autoencoder on the input. Furthermore, Kwon et al. [46] proposes an asymmetric reverse process that discovers the semantic latent space of a frozen diffusion model, where modification in the space synthesizes various attributes on input images. However, in principle, diffusion models lack a semantic latent space and it is unclear how to efficiently learn representations using their capabilities.

2.2. Causal Representation Learning

Given the inherent challenges of identifiability in causal representation learning, many previous studies have tackled this issue by imposing certain assumptions on the dataset or the causal structure. Several previous methods rely on additional knowledge of the data generation process, such as knowledge of the causal graph or labels for high-level causal variables. CausalGAN [55] requires the structure of the underlying causal graph to be known. Yang et al. [11] and Liu et al. [12] assume a linear structural equation model, and they require additional information associated with the true causal concepts as supervising signals. Similar to Yang et al. [11], Komanduri et al. [56] assumes the availability of supplementary supervision labels but without requiring mutual independence among factors. Von Kügelgen et al. [57] investigates self-supervised causal representation learning by utilizing a known, but non-trivial, causal graph between content and style factors. Subramanian et al. [13] applies Bayesian structure learning in the latent space and relies on having interventional samples. Sturma et al. [58] considers a setup where the authors have access to data from multiple domains that share a causal representation. Buchholz et al. [59] assumes the latent distribution is Gaussian and the authors have access to unknown single-node interventional samples. Additionally, Ahuja et al. [15] analyzes various scenarios and the level of identifiability in the presence of interventional data. For an overview of causal representation learning, we refer to Schölkopf et al. [1].

Furthermore, there have been recent works on utilizing diffusion models in causality. Specifically, Sanchez and Tsaftaris [60] focuses on counterfactual estimation from observational imaging data given a known causal structure. Similarly, Sanchez et al. [61] aims to learn the underlying SCM in the low-level data space assuming a non-linear additive noise model, which is identifiable. However, both of these works focus on the SCM in the data space, while our approach focuses on learning the SCM in the latent space among the underlying latent variables in a weakly supervised setting. Other relevant work closely related to causal representation learning includes disentangled representations and independent component analysis [62–66].

3. Structural Causal Model

Following refs. Pearl [67], Bongers et al. [68], we describe the data-generating process (DGP) using the notion of structural causal models. A structural causal model (SCM) is a formal framework used to represent and analyze causal relationships among variables within a system. An SCM essentially consists of a set of random variables, and measurable functions between them specifying the underlying causal relationships of the DGP. We formally define SCMs as follows.

Definition 1 (Structural Causal Model (SCM), Definition 2.1 by Bongers et al. [68])**.** *A structural causal model (SCM) is a tuple $\langle \mathtt{L}, \mathtt{J}, \mathcal{E}, \mathcal{Z}, f, \mu \rangle$, where (i) $\mathtt{L}$ is a finite index set of endogenous variables; (ii) $\mathtt{J}$ is an index set of exogenous variables, which is disjoint with $\mathtt{L}$; (iii) $\mathcal{E} = \prod_{j \in \mathtt{J}} \mathcal{E}_j$ is the product of the domains of the exogenous variables, where each $\mathcal{E}_j$ is a measurable space; (iv) $\mathcal{Z} = \prod_{j \in \mathtt{L}} \mathcal{Z}_j$ is the product of the domains of the endogenous variables, where each $\mathcal{Z}_j$ is a measurable space; (v) $f : \mathcal{Z} \times \mathcal{E} \to \mathcal{Z}$ is a measurable function that specifies the causal mechanism; and (vi) $\mu = \prod_{j \in \mathtt{J}} \mu_j$ is a product measure, where μ_j is a probability measure on $\mathcal{E}_j$ for each $j \in \mathtt{J}$.*

In the definition above, the functional relationships between variables are expressed in terms of a function f. This feature allows us to model the cause–effect relationships of the data-generating process (DGP) using *structural equations*. Structural equations are mathematical representations used to describe causal relationships among variables in a system. They express how one or more variables causally influence others within a causal graphical model. For a given SCM as above, a structural equation specifies an endogenous random variable z_l via a measurable function of the form $z_l = f_l(\boldsymbol{z}, \boldsymbol{e})$ where $\boldsymbol{z} \in \mathcal{Z}, \boldsymbol{e} \in \mathcal{E}$. This function essentially captures the deterministic relationships specified by

f as in Definition 1. A *parent* $i \in \mathtt{L} \cup \mathtt{J}$ of l is any index for which there is no measurable function $k\colon \prod_{j \in \mathtt{L} \setminus \{i\}} \mathcal{Z}_j \times \mathcal{E} \to \mathcal{Z}_l$ with $f_l = k$ almost surely. Intuitively, each endogenous variable z_l is specified by its parents together with the exogenous variables via the structural equations.

A structural equation model as in Definition 1 can be conveniently described with the *causal graph*, a directed graph of the form $\mathcal{G} = (V, E)$. The nodes of the causal graph consist of the entire set of indices for the endogenous variables, and the edges are specified by the structural equations, i.e., $\{j \to l\} \in E$ if and only if j is a parent of l. Note that the variables in the set $\mathtt{pa}(z_l)$ are indexed by the parent nodes of l in the corresponding graph $\mathcal{G}$. An example of a causal diagram is given in Figure 1(left).

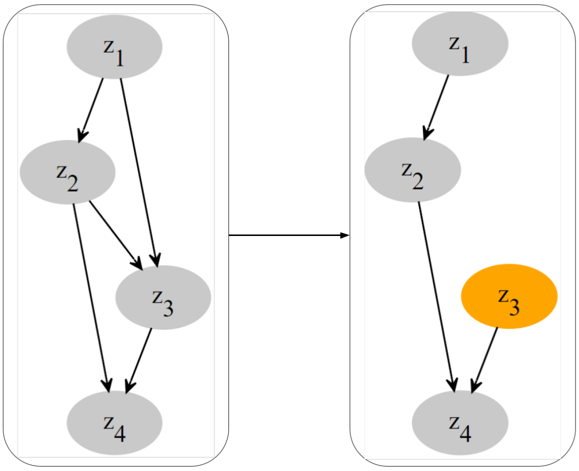

Figure 1. A causal graph before and after an intervention. Applying a perfect intervention on z_3 eliminates the dependencies between this node and its parents in the causal graph.

Solution Functions. An alternative way of defining SCMs replaces causal mechanisms with solution functions $h : \mathcal{E} \to \mathcal{Z}$ which maps exogenous noise variables to endogenous causal variables, i.e., $z_i = h_i(e), e \in \mathcal{E}$, and is defined by successively applying the causal mechanisms f. Solution functions contain the same information as causal mechanisms and they can be derived from each other. We utilize this formulation in our framework.

Interventions. A very important aspect of SCMs is that they allow us to reason about cause–effect relationships using *interventions*. Interventions refer to deliberate changes or manipulations made to one or more variables within the model to study their causal effects on other variables. In this paper, we specifically consider perfect interventions [67]. For a given SCM as in Definition 1, consider a variable $\boldsymbol{W} := \prod_{j \in \mathtt{L}'} \mathcal{Z}_j$ for a set $\mathtt{L}' \subseteq \mathtt{L}$, and let $\boldsymbol{w} := \prod_{j \in \mathtt{L}'} w_j$ be a point of its domain. The perfect intervention $\boldsymbol{W} \leftarrow w$ amounts to replacing the structural equations $z_j = f_j(z, e)$ with the constant functions $z_j \equiv w_j$ for all $j \in \mathtt{L}'$. We denote with $z \mid do(w)$ the variables z after performing the interventions. This procedure defines a new probability distribution $p_z(z \mid do(w))$, which we refer to as interventional distribution. This distribution entails the following information: If we apply $do(\boldsymbol{w})$, what will be the value of z? We extend this definition by defining I as the set of interventions entailed by w, and we utilize this formulation in our framework. An example of a causal graph and a single perfect intervention is depicted in Figure 1.

Equivalence of SCMs. We now define the concept of equivalence between structural causal models. Two SCMs are structurally equivalent if their respective sets of structural equations and exogenous variables are equivalent. Formally, the notion of equivalence is defined as follows.

Definition 2. *Consider two SCMs $\langle \mathtt{L}, \mathtt{J}, \mathcal{E}, \mathcal{Z}, f, \mu \rangle$ and $\langle \mathtt{L}', \mathtt{J}', \mathcal{E}', \mathcal{Z}', f', \mu' \rangle$. Consider their respective causal graphs $\mathcal{G}$ and $\mathcal{G}'$. An isomorphism between the two SCMs consists of the following:*

(A) A graph isomorphism $\sigma : \mathcal{G} \to \mathcal{G}'$; (A graph isomorphism $\sigma : \mathcal{G} \to \mathcal{G}'$ is a bijective map from the vertices of $\mathcal{G}$ to the vertices of $\mathcal{G}'$, such that there exist an edge $\sigma(i) \to \sigma(j)$ in $\mathcal{G}'$ iff. there exist an edge $i \to j$ in $\mathcal{G}$.)

(B) Measure-preserving (A measure-preserving function $l : \boldsymbol{A} \to \boldsymbol{B}$ ensures that the probability distribution in the domain space A remains the same when mapped to the co-domain space B through the function l.) invertible functions $l_j : \mathcal{Z}_j \to \mathcal{Z}'_{\sigma(j)}$ such that the function $l(z) := \prod_{j \in \mathrm{L}} l_j(z_j)$ yields $f'(l(z), e) = l(f(z, e))$ for all $z \in \mathcal{Z}, e \in \mathcal{E}$.

We say that two SCMs are equivalent if their domains are identical and such an isomorphism exists between them.

Definition 2 ensures that the causal mechanisms of equivalent SCMs are essentially identical. The functions l_j in Definition 2 reparameterize the random variables in both models such that the structural equations and causal relationships are preserved.

4. Diffusion Models

4.1. Overview

The fundamental concept behind diffusion-based generative models is to learn to generate data by inverting the diffusion process. Diffusion models comprise two processes: a forward process and a backward process. The forward process gradually adds noise to data and maps data to (almost) pure noise. The backward process, on the other hand, is used to go from a noise sample back to the original data space.

The forward process is defined by a stochastic differential equation (SDE) across a continuous time domain $t \in [0, 1]$, aiming to transform the data distribution to a known prior distribution, typically a standard multivariate Gaussian. Given x_0 sampled from a data distribution $p(x_0)$, the forward process constructs a trajectory $(x_t)_{t \in [0,1]}$ across the time domain. We utilize the Variance Exploding SDE [53] for the forward process, which is defined as:

$$dx = f(x, t) + g(t)dw := \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw,$$

where w is the standard Wiener process, and $\sigma^2(t)$ is the noise variance of the diffusion process at time t. The backward process is also formulated as an SDE in the following manner:

$$dx = [f(x, t) - g^2(t)\nabla_x \log p_t(x)]dt + g(t)d\bar{w}\,,$$

where $\bar{w}$ is the standard Wiener process in reverse time.

Score matching. To use this backward process, the score function $\nabla_x \log p_t(x)$ is required. It is usually approximated by a neural score function $s_\theta(\cdot)$ which can be trained by Explicit Score Matching [69] defined as:

$$\mathcal{L}(\theta) = \mathbb{E}_t \left[\lambda(t) \mathbb{E}_{p(x_t)} \left[||s_\theta(x_t, t) - \nabla_{x_t} \log p_t(x_t)||^2 \right] \right],$$

where $\lambda(t)$ is a positive weighting function. However, the ground-truth score function $\nabla_x \log p_t(x)$ is generally not known. Vincent [70] addresses this issue by proposing Denoising Score Matching. The approximate score function is then learned by minimizing the loss function:

$$\mathcal{L}(\theta) = \left[\lambda(t) \mathbb{E}_{x_0} \mathbb{E}_{p(x_t|x_0)} \left[||s_\theta(x_t, t) - \nabla_{x_t} \log p_t(x_t|x_0)||^2 \right] \right],$$

where the conditional distribution of x_t given x_0 is $p_t(x_t|x_0) = \mathcal{N}(x_t; x_0, [\sigma^2(t) - \sigma^2(0)]\boldsymbol{I})$. This objective function originates from the Evidence Lower Bound (ELBO) of the data distribution, and it has been shown that with a specific weighting function, this objective function becomes exactly a term in the ELBO [53]. For more details, see Appendix B.

4.2. Diffusion-Based Representations

Conditional Score Matching. We can modify Denoising Score Matching so that the score function receives additional information through an external trainable module. This results in a conditional diffusion model which allows to perform representation learning while training the score function. Abstreiter et al. [43] proposes conditional Denoising Score Matching defined as:

$$\mathcal{L}(\theta,\phi) = \mathbb{E}_t\left[\lambda(t)\mathbb{E}_{x_0}\mathbb{E}_{p(x_t|x_0)}\left[||s_\theta(x_t, E_\phi(x_0), t) - \nabla_{x_t}\log p_t(x_t|x_0)||^2\right]\right], \tag{1}$$

where the score function is conditioned on a module $E_\phi(x_0)$ which provides additional information about the data to the diffusion model through a learned encoder with parameters ϕ. In fact, the encoder learns to extract necessary information from x_0 in a reduced-dimensional space that helps recover x_0 by denoising x_t. Abstreiter et al. [43] also presents an alternative objective where the encoder is a function of time. Formally, the new objective is

$$\mathcal{L}(\theta,\phi) = \mathbb{E}_t\left[\lambda(t)\mathbb{E}_{x_0}\mathbb{E}_{p(x_t|x_0)}\left[||s_\theta(x_t, E_\phi(x_0, t), t) - \nabla_{x_t}\log p_t(x_t|x_0)||^2\right]\right], \tag{2}$$

With this objective, the encoder learns a representation trajectory of x_0 instead of a single representation. Training this system has the potential to minimize the objective to zero, motivating the encoder $E_\phi(.)$ to learn meaningful, distinct representations at different timesteps [43,44].

Comparison with Other Generative Models. The key difference between the other generative models and diffusion-based representations is that other generative models are only concerned with one finite code and all the information is encoded into this single code, while in the latter, different levels of information are encoded along an infinite-dimensional code, i.e., the encoder is conditioned on time t and produces a trajectory-based representation $(E_\phi(x_0, t))_{t\in[0,1]}$. Within this representation, various points along the trajectory contain different levels of information as highlighted by Mittal et al. [44]. In this work, we first explore a time-independent single code, where we employ Equation (1) and show that with a certain weighting function, this objective function will become the ELBO. Then, we apply the same experiments with infinite-dimensional latent code (Equation (2)) and study the benefits and implications of these formulations for causal representation learning.

5. Problem Formulation

We consider a system that is described by an unknown underlying SCM on the latent causal variable z, where we have access to low-level data pairs $(x_0, \tilde{x}_0) \sim p(x_0, \tilde{x}_0)$ representing the system before and after a random, unknown, and atomic intervention. We consider the assumptions and the data-generation process that will be described in Section 5.1. Our objective is to learn an SCM that accurately represents the true underlying SCM associated with the given data, up to a permutation and elementwise reparameterization of causal variables and solution functions. To this end, we train an SCM by maximizing the likelihood of data. With sufficient data and perfect optimization, we can find the SCM that is equivalent to the ground-truth SCM.

5.1. Weakly Supervised Framework

We build our weakly supervised framework on the assumptions and identifiability conditions established by Brehmer et al. [14]. We try to learn the underlying SCM over unknown latent causal variables z of a system in which low-level information $x_0 \in \mathcal{X}$ generated directly from z through an unknown function $g : \mathcal{Z} \to \mathcal{X}$ is available. Following Brehmer et al. [14], Locatello et al. [26], we consider a dataset that consists of paired datapoints $(x_0, \tilde{x}_0)$, generated as follows:

$$\begin{aligned} &\boldsymbol{e} \sim p_e(\boldsymbol{e}), \quad \boldsymbol{I} \sim p_I(\boldsymbol{I}), && \boldsymbol{z} = h(\boldsymbol{e}), \quad \boldsymbol{x_0} = g(\boldsymbol{z}) \\ &\tilde{\boldsymbol{e}} \sim p_e(\boldsymbol{e} \mid do(\boldsymbol{e'})) \text{ with } \boldsymbol{e'} \sim p_{e_I}(\boldsymbol{e'}), && \tilde{\boldsymbol{z}} = \tilde{h}_I(\tilde{\boldsymbol{e}}), \quad \tilde{\boldsymbol{x}}_\mathbf{0} = g(\tilde{\boldsymbol{z}}) \end{aligned}$$

where e and $\tilde{e}$ are the exogenous noise variables of the underlying SCM, $h(\cdot)$ and $\tilde{h}_I(\cdot)$ are the solution functions before and after a single perfect intervention I, and $p_I(\cdot)$ is a prior on all possible values of atomic interventions such that $p_{e_I}(e') > 0$ for every possible atomic intervention. In this setting, $p_e(\boldsymbol{e} \mid do(e'))$ is defined such that the noise variable remains the same and changes only for the element that is intervened upon, i.e., $\tilde{\boldsymbol{e}}_I = e' \neq \boldsymbol{e}_I$, $\tilde{\boldsymbol{e}}_{\setminus I} = \boldsymbol{e}_{\setminus I}$. Since the intervention is perfect, the solution function will also change in a way that only for the intervened variable is the dependency between the latent causal variable z_I and its parents removed. For the complete list of assumptions, see Appendix A.

It is proven that under this weakly supervised setting, it is possible to identify the latent causal variables and solution functions up to a permutation and elementwise reparameterization of the variables. For the proof of the identifiability of the described system, we refer to Brehmer et al. [14].

5.2. Non-Identifiability from Observational Data

In this section, we show that interventions are necessary for identifiability in this setting. In fact, note that Definition 2 implies that the distributions of two equivalent SCMs are the same, up to a measure-preserving invertible function. However, two SCMs may entail the same observational distribution on the generated data, even if their respective causal mechanisms are not equivalent. This is best illustrated with an example. Consider two datasets $\{X_1, Y_1\}$ and $\{X_2, Y_2\}$. The respective DGPs are:

$$X_1, Y_1 \sim \mathcal{N}(\mathbf{0}, \Sigma) \quad \text{and} \quad \begin{cases} X_2 \sim \mathcal{N}(0, 1) \\ Y_2 \sim X_2 \end{cases}$$

where the covariance matrix Σ is defined as

$$\Sigma = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}.$$

Note that both datasets $\{X_1, Y_1\}$ and $\{X_2, Y_2\}$ entail the same observational distribution. However, these datasets have different causal mechanisms. In particular, their respective causal diagrams are not isomorphic. Hence, by this, we see that the same observational distribution may entail different causal diagrams. This means that the causal dynamics of an SCM cannot be inferred from the distribution of a given observational dataset, i.e., SCMs are unidentifiable from observational data.

5.3. Limitations

While our goal is to execute a robust and informative study to address the selected research question, it is important to acknowledge inherent limitations related to data, model assumptions, and evaluations. First, our evaluation is limited to synthetic datasets in a single modality. Furthermore, we consider the weakly supervised data-generation process and assumptions for the identifiability of the underlying model, which may limit the practical application of our work in systems where the assumptions do not hold. Finally, the representation learning process relies on an encoder, which acts as an information channel, regulating the amount of input information transmitted to the score function during each step of the diffusion process. It is important to note that in certain scenarios, the encoder may not be essential to the diffusion process and could potentially result in collapsing behavior. However, it is important to emphasize that our work is a preliminary step towards utilizing diffusion models for causal representation learning and lays the foundation for significant further research in this area.

6. The DCRL Framework

Figure 2 provides a visual representation of the framework's architecture. In this study, we utilize a conditional diffusion model and apply it to the input data $(x_0, \tilde{x}_0)$, where $x_0, \tilde{x}_0 \in \mathbb{R}^{3\times W\times H}$ and W and H are the width and height of the input, respectively. We denote $(x_t)_{t\in[0,1]}$ as the diffusion trajectory across the time domain with x_0 as the input data. The conditioning module is defined as the encoding module, generating high-level diffusion-based representations $(e, \tilde{e})$ for each low-level data pair, where $e, \tilde{e} \in \mathbb{R}^d$ and d is the number of latent causal variables assumed to be known. We empirically show that these latent variables contain equivalent information as in noise variables of the underlying SCM and can be used interchangeably. Then, we infer the intervention target $I \in \{0, 1, ..., d-1\}$ for each data pair by an intervention module and use neural solution functions on top of the latent variables $(e, \tilde{e})$ and the intervention target I to obtain the underlying latent causal variables $z, \tilde{z} \in R^d$. We base our framework on the Implicit Latent Causal Model (ILCM) introduced by Brehmer et al. [14] and describe each part of our framework in the next paragraphs.

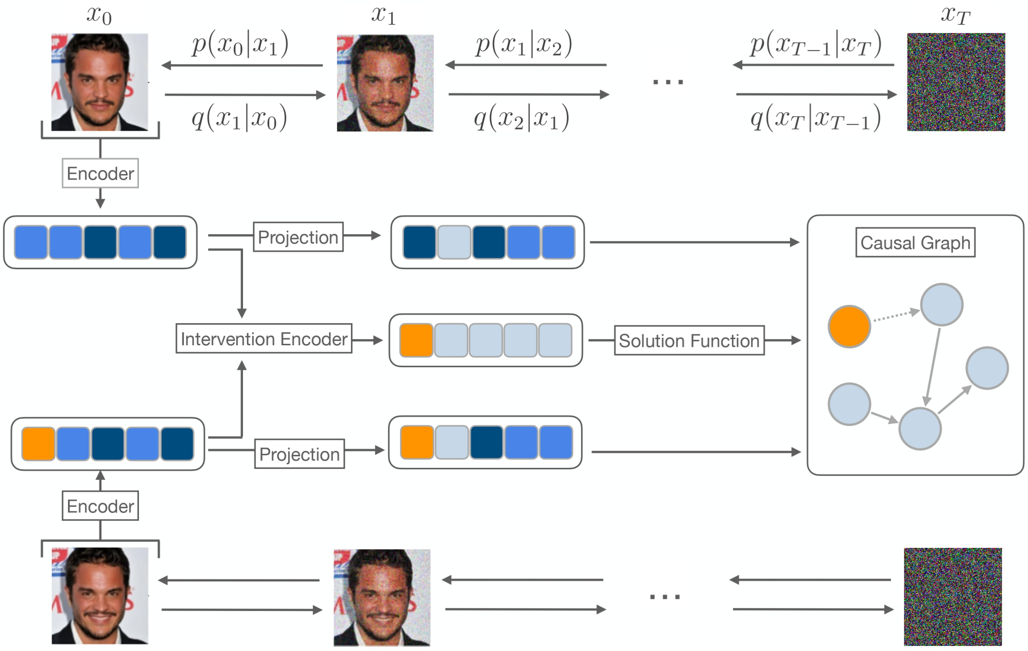

Figure 2. Overview of our framework. Here, we have a paired image of a face before and after an intervention (the smile). The paired image is mapped to latent variables by a stochastic encoder. The intervention target is determined by applying the intervention encoder to these latent variables. To maintain the weakly supervised structure, the latent variables are projected into a new pair and then serve as the conditioning module for a conditional diffusion model. The projected latent variables are in fact diffusion-based representations of the input pair. Finally, they are utilized in neural solution functions together with the intervention target to obtain the latent causal variables.

6.1. Conditional Diffusion Model

Based on the formulation described in Section 4, we use a conditional diffusion model. A stochastic encoder $q(e|x_0)$ serves as the conditioning module, mapping low-level data space to high-level latent space. When employing a finite code where the stochastic encoder is independent of time, e is a single vector of size d. In this case, the framework learns a single SCM. Alternatively, in the case of using infinite-dimensional latent code, the stochastic encoder generates $(e_t)_{t\in[0,1]}$ which is a trajectory-based representation across time. At each timestep t, $e_t \in \mathbb{R}^d$ represents a single point of the trajectory. In this scenario, the framework learns an SCM at each timestep. In the following paragraphs, for the sake of simplicity, we use the single-code formulation.

6.2. The Encoding and the Intervention Module

The encoding module consists of two main parts: the *stochastic encoder* and the *projection module*. The stochastic encoder $q(e|x_0)$ maps data pairs $(x_0, \tilde{x}_0)$ to pre-projection latent variables $(e, \tilde{e})$. The encoded inputs are then utilized in the intervention module $q(I|x_0, \tilde{x}_0)$ to infer the intervention target I for the data pair $(x_0, \tilde{x}_0)$. Based on our data generation process in Section 5.1, the encoded inputs have the property that only for the elements that are intervened upon do we have $e_i \neq \tilde{e}_i, i \in I$, and the rest will remain the same. Based on this property, in order to infer interventions, we employ an intervention module $q(I|x, \tilde{x})$ which is defined heuristically as

$$q(i \in I|x_0, \tilde{x}_0) = \frac{1}{Z}(\alpha + \beta|\mu_e(x_0)_i - \mu_e(\tilde{x}_0)_i| + \gamma|\mu_e(x_0)_i - \mu_e(\tilde{x}_0)_i|^2)$$

where $\mu_e(x_0)$ is the mean of the stochastic encoder $q(e|x_0)$; α, β, and γ are learnable parameters; and Z is a normalization constant. This simple heuristic function ensures that a variable has a higher chance to be selected as the intervened variable if it undergoes more significant changes in response to the intervention. Once the intervention is inferred from the pre-projection latent variables, we apply the projection module. Similar to Brehmer et al. [14], the projection module is dependent on the inferred intervention target I and projects the encoded input $(e, \tilde{e})$ to new latent variables in a way that for the components e_i that are not intervened upon $i \notin I$, the pre-intervention and post-intervention latent components will be equal $e_i = \tilde{e}_i$. This prevents the framework from deviating from the weakly supervised structure.

We write the combination of the encoder and the projection module as $q(e, \tilde{e}|x_0, \tilde{x}_0, I)$, and refer to it as the *encoding module*. By this definition, the encoding module $q(e, \tilde{e}|x_0, \tilde{x}_0, I)$ maps the input $(x_0, \tilde{x}_0)$ to latent variables $(e, \tilde{e})$ and the intervention module infers the intervention I based on pre-projection latent variables.

6.3. Prior

Given the intervention target I and latent variables $(e, \tilde{e})$, we define the prior $p(e, \tilde{e}, I)$ as $p(e, \tilde{e}, I) = p(I)p(e)p(\tilde{e}|e, I)$. The objective of the prior distribution is to implicitly capture the causal structure and causal mechanisms within the system. Specifically, $p(I)$ and $p(e)$ denote the prior distributions over intervention targets and latent variables, respectively, and are configured as uniform categorical with each latent variable as a category, and standard Gaussian distributions, respectively. According to our data generation process, when an intervention is applied, only the elements in the latent variables that are intervened upon are altered; the other elements remain unchanged and independent of each other. Consequently, we can define $p(\tilde{e}|e, I)$ as follows:

$$p(\tilde{e}|e, I) = \prod_{i \notin I} \delta(\tilde{e}_i - e_i) \prod_{i \in I} p(\tilde{e}_i|e)$$

In this equation, $\delta(.)$ is the Dirac delta function that fulfills this property for non-intervened latent variables.

6.4. Neural Solution Functions

In order to encode the information about the intervened variables, we incorporate a conditional normalizing flow $p(\tilde{e}_i|e)$ defined as

$$p(\tilde{e}_i|e) = \tilde{p}(h_i(\tilde{e}_i; e_i)) \left| \frac{\partial h_i(\tilde{e}_i; e_i)}{\partial \tilde{e}_i} \right|$$

where $h(.)$ are the solution functions of the SCM. They are defined as invertible affine transformations with parameters learned with neural networks. Therefore, by learning solution functions, i.e., learning to transform e to z, we implicitly model the causal graph into the framework and obtain the latent causal variables. For more details about the implementation, see Appendix C.

6.5. The Evidence Lower Bound for DCRL

We calculate the Evidence Lower Bound (ELBO) for the proposed model for the framework described in the previous section. In the case of having single-point representations in which the noise variable e is independent of time, the ELBO becomes:

$$\begin{aligned}\mathcal{L}_{model} = \mathbb{E}_{p(x_0,\tilde{x}_0)}\mathbb{E}_{q(I|x_0,\tilde{x}_0)}\mathbb{E}_{q(e,\tilde{e}|x_0,\tilde{x}_0,I)}\mathbb{E}_{t\sim U(0,1)}\mathbb{E}_{q(x_t|x_0)}\mathbb{E}_{q(\tilde{x}_t|\tilde{x}_0)}\Bigg[\lambda(t)||s_\theta(x_t,e,t) \\ - \nabla_{x_t}\log p(x_t|x_0)||_2^2 + \lambda(t)||s_\theta(\tilde{x}_t,\tilde{e},t) - \nabla_{\tilde{x}_t}\log p(\tilde{x}_t|\tilde{x}_0)||_2^2 + \beta\Big[\log p(I) + \log p(e) \\ + \log p(\tilde{e}|e,I) - \log q(I|x_0,\tilde{x}_0) - \log q(e,\tilde{e}|x_0,\tilde{x}_0,I)\Big]\Bigg],\end{aligned}$$

where $\lambda(t)$ is a positive weighting function, and $\beta = 1$. We train the model by minimizing a reweighted loss function reminiscent of β-VAEs, setting β to 0 and increasing it to 1 during training.

In the case of using infinite-dimensional representations (Equation (2)), the objective function becomes:

$$\begin{aligned}\mathcal{L}_{model} = \mathbb{E}_{p(x_0,\tilde{x}_0)}\mathbb{E}_{q(I|x_0,\tilde{x}_0)}\mathbb{E}_{t\sim U(0,1)}\mathbb{E}_{q(e_t,\tilde{e}_t|x_0,\tilde{x}_0,I)}\mathbb{E}_{q(x_t|x_0)}\mathbb{E}_{q(\tilde{x}_t|\tilde{x}_0)}\Bigg[\lambda(t)||s_\theta(x_t,e_t,t) \\ - \nabla_{x_t}\log p(x_t|x_0)||_2^2 + \lambda(t)||s_\theta(\tilde{x}_t,\tilde{e}_t,t) - \nabla_{\tilde{x}_t}\log p(\tilde{x}_t|\tilde{x}_0)||_2^2 + \beta\Big[\log p(I) + \log p(e_t) \\ + \log p(\tilde{e}_t|e_t,I) - \log q(I|x_0,\tilde{x}_0) - \log q(e_t,\tilde{e}_t|x_0,\tilde{x}_0,I)\Big]\Bigg], \qquad (3)\end{aligned}$$

where $(e_t)_{t\in[0,1]}$ is the trajectory-based representation and $e_t \in \mathbb{R}^d$ is the single point of the trajectory at time t. For a complete derivation of the ELBO, see Appendix B.

To prevent a collapse of the latent space to a lower-dimensional subspace, we add the negative entropy of the batch-aggregate intervention posterior as a regularization term to the loss function:

$$\mathcal{L}_{entropy} = \mathbb{E}_{batches}\Big[-\sum_I q_I^{batch}(I)\log q_I^{batch}(I)\Big]$$

where $\mathbb{E}_{batches}[\,\cdot\,]$ is the expected value over all the batches of data, and $q_I^{batch}(I)$ is defined as

$$q_I^{batch}(I) = \mathbb{E}_{x_0,\tilde{x}_0\in batch}[q(I|x_0,\tilde{x}_0)]$$

After the training, the framework contains information about the underlying causal structure and latent causal variables, and it can be used in different downstream tasks.

7. Experiments

Here, we analyze the performance of the proposed model, DCRL, on synthetic data. We employ DCRL for the task of causal discovery. After training DCRL, we use the framework to obtain causal variables $(z,\tilde{z})$ for the test set, and apply ENCO [71], a continuous optimization structure learning method that leverages observational and interventional data, on the obtained samples to infer the underlying causal graph. Furthermore, we evaluate the learned causal variables with the DCI framework [72].

Data Generation. In order to generate latent causal variables, we adopt random graphs, where each edge in a fixed topological order is sampled from a Bernoulli distribution with a parameter that is equal to 0.5. We consider the SCM to be linear Gaussian and we sample the weights from a multivariate normal distribution with zero mean and unit variance. We make sure the weights are not close to zero to avoid violation of the faithfulness assumption. We introduce additive Gaussian noise with equal variances across all nodes, with its variance set to 0.1. Latent causal variables are then sampled using ancestral sampling, and we generate 10^5 training samples, 10^4 validation samples, and 10^4 test samples. Finally, to generate input data x_0, we apply a random linear projection on the obtained latent variables. We keep the dimension of x_0 fixed to 16. We utilize an SCM with 5, 10, and 15 variables. To enhance the robustness of the results, we generate data for 4 different seeds and repeat our experiments for each seed.

Baselines. We consider ILCM [14] as our main baseline. To the best of our knowledge, there are no other methods that consider the same weakly supervised assumptions, and adapting other methods to our assumptions either substantially changes the method or is infeasible. We also evaluate the outcomes against a variation of disentanglement VAE proposed by [26] tailored for weakly supervised settings. This model, referred to as d-VAE, models the weakly supervised process but assumes unconnected variation factors instead of a causal relationship among variables. Similarly, we apply ENCO on top of both to obtain the learned graph.

Metrics. We assess the performance of models with the following metrics:

- The *Structural Hamming Distance (SHD)* is a metric used to quantify the dissimilarity between two directed acyclic graphs (DAGs) by measuring the minimum number of edge additions, deletions, and reversals required to transform one graph into another. It is calculated by summing up the absolute differences between the entries of adjacency matrices of two graphs.
- The *DCI Disentanglement Score* is a metric used to evaluate the disentanglement quality of a generative model and takes values between 0 and 1. Disentanglement refers to the extent to which the model learns to predict the underlying factors of variation in the data in a way that each predicted variable captures at most one underlying factor. If a predicted factor is important to predict a single underlying factor, the score will be 1, and if a predicted factor is equally important to predict all the underlying factors, the score will be 0 [72].
- The *DCI Completeness Score* measures how well each underlying factor of variation is captured by a single predicted latent variable and has a value between 0 and 1. If a single variable contributes to one underlying factor, the score will be 1, and if all variables equally contribute to the prediction of a single factor, the score will be 0 [72].

7.1. Single-Point Representations

Utilizing single-point representations where $e \in \mathbb{R}^d$ and is independent of time, our method demonstrates superior or competitive performance compared to the baselines as indicated by the metrics shown in Figure 3. The d-VAE performs poorly across all metrics primarily because it assumes independent rather than causal relationships among variables. In scenarios involving 5 and 10 causal variables, ILCM shows comparable performance to DCRL, suggesting that a standard VAE can sufficiently capture essential information about causal factors. However, in higher dimensions, our method excels by capturing more detailed information about causal variables and their underlying structure. Our findings indicate that diffusion-based representations are more beneficial in higher dimensions, providing more accurate information about the underlying causal variables compared to other baseline methods.

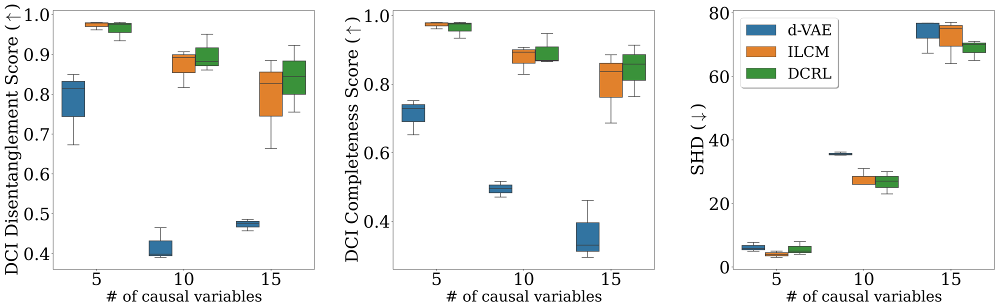

Figure 3. Comparison of models on different metrics when using single-point representations. Our approach outperforms or competes favorably with the baseline methods on all metrics. Particularly in higher dimensions, our method excels by capturing additional information about the causal variables and the underlying causal structure.

7.2. *Infinite-Dimensional Representations*

In these experiments, we utilize the infinite-dimensional representations approach to develop trajectory-based representations for each input x_0, denoted as $(e_t)_{t\in[0,1]}$. In order to perform inference, we sample points from this trajectory at intervals of 0.1 resulting in 11 specific timesteps. The outcomes are depicted in Figure 4. Generally, representations in the middle of the trajectory contain the most information and are comparable to or even outperform the baselines. Going further in time, representations appear to lose information but improve as they move towards the end of the trajectory. This phenomenon arises because during training, as we are further in time, the noise in the diffusion model is fairly high and the conditioning module compensates for that by providing the necessary information for the diffusion model to learn the score function.

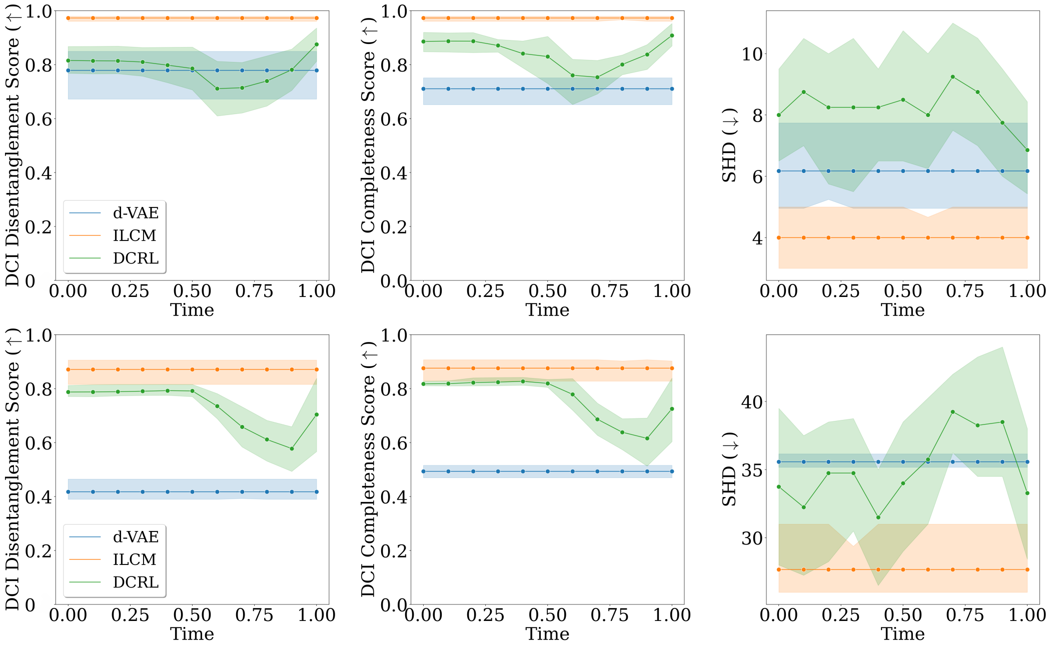

Figure 4. *Cont.*

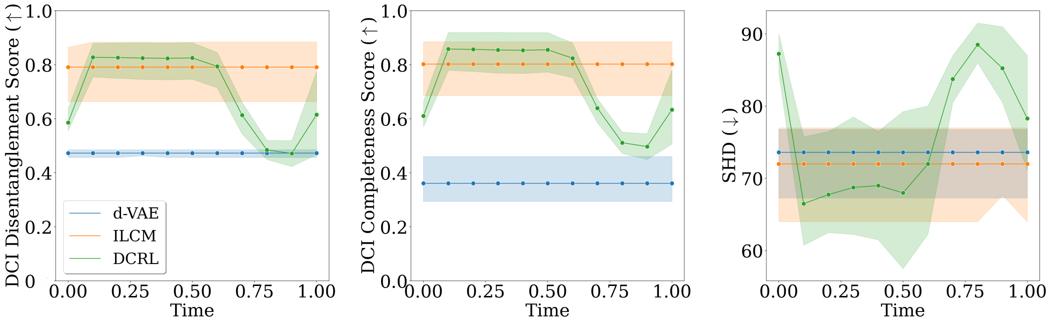

Figure 4. Comparison of models on different metrics when using infinite-dimensional representations. From top to bottom, each row corresponds to experiments with 5, 10, and 15 causal variables, respectively. We sample points from the trajectory at intervals of 0.1, creating a total of 11 specific timesteps. Typically, representations in the middle of the trajectory carry the most information, often matching or surpassing the baseline performance. As we move further in time, representations seem to lose some information, but they improve as they approach the end of the trajectory. Furthermore, the framework performs worse or on par with baselines in lower dimensions but generally outperforms them in higher dimensions.

8. Conclusions

Identifying the underlying causal variables and mechanisms of a system solely from observational data is considered impossible without additional assumptions. In this project, we use weak supervision as an inductive bias and study whether the information encoded in the latent code of diffusion-based representations contains useful knowledge of causal variables and the underlying causal graph.

This study represents an initial exploration of applying diffusion models to causal representation learning, highlighting the need for further research and extensions in this area. Our method relies on an external encoder to provide necessary information for the diffusion model to learn the score function. Future work could focus on integrating more efficient ways of acquiring representations from diffusion models without external dependencies or conditioning. Additionally, extending the weakly supervised framework to higher dimensions and other modalities, such as video or multi-view data, is another potential direction. Applying the proposed method to domains such as experimental design, reinforcement learning, and robotics—where the independent actions can be considered interventions and the system's state before and after an action is observable—presents another promising avenue for research. Finally, extending the framework to other settings, such as dynamical systems, where the infinite-dimensional latent code corresponds to the system's state at different timesteps, is another interesting potential direction.

Author Contributions: Conceptualization, A.M.K.M., A.D., S.B. and F.Q.; Methodology, A.M.K.M., A.D., S.B. and F.Q.; Software, A.M.K.M.; Validation, A.M.K.M., A.D. and F.Q.; Investigation, A.M.K.M., A.D. and F.Q.; Resources, S.B. and K.H.J.; Writing—original draft preparation, A.M.K.M., A.D., S.B. and F.Q.; Writing—review and editing, A.M.K.M., A.D., S.B. and F.Q.; Supervision, S.B., K.H.J. and F.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not Applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by Knut and Alice Wallenberg Foundation, and the computations were enabled by the Berzelius resource provided by the Knut and Alice Wallenberg Foundation at the National Supercomputer Centre.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Identifiability Conditions

We give all the assumptions used by [14] for the SCM to be identifiable up to a permutation and elementwise reparameterization of the causal variables. These assumptions are as follows:

- **Causal Sufficiency** (see Definition 6.2.2 by Pearl [67]). All the causal variables are measurable, and the noise variables are mutually independent.
- **Faithfulness** (see Definition 2.4.1 by Pearl [67]). All the independencies of data distribution are encoded in the graph.
- **Acyclicity.** The ground-truth graph is acyclic.
- **Diffeomorphic Causal Mechanisms** (see Brehmer et al. [14]). We require causal mechanisms and, therefore, solution functions $h(\cdot)$ to be diffeomorphic, that is, for any possible input value of the causal mechanisms $f(\cdot)$, f is invertible, differentiable, and its inverse is differentiable.
- **Observability of All Interventions** (see Brehmer et al. [14]). The intervention distribution $p_I(\cdot)$ has support for any atomic intervention, i.e., $p_I(z_j) > 0, \quad \forall j \in L$. In other words, the dataset contains data pairs generated from interventions on any causal variable.
- **Perfect Atomic Interventions** (see Section 3.2. by Pearl [67], and Brehmer et al. [14]). We assume that the intervention set I contains atomic interventions on causal variables in which the intervention is perfect, i.e., the intervened-upon mechanism is independent of any causal variable.

Appendix B. Problem Formulation and ELBO Derivation

Here, we derive the ELBO for the proposed framework. To avoid confusion, in the notation, we separate the latent variables of the diffusion model and the input data. We denote them with u and x_0, respectively. Furthermore, for simplicity, we only derive the ELBO when using single-point representations independent of time, i.e., $e \in \mathbb{R}^d$. The ELBO for the infinite-dimensional can be derived similarly. The ELBO for the framework is calculated as:

$$
\begin{aligned}
&\log\ p(x_0, \tilde{x}_0) \geq \mathbb{E}_{q(e,\tilde{e},u,\tilde{u},I|x_0,\tilde{x}_0)}\left[\log \frac{p(x_0, \tilde{x}_0, u, \tilde{u}, e, \tilde{e}, I)}{q(e, \tilde{e}, I, u, \tilde{u}|x_0, \tilde{x}_0)}\right] \\
&=\mathbb{E}_{q(e,\tilde{e},u,\tilde{u},I|x_0,\tilde{x}_0)}\left[\log \frac{p(I)}{q(I|x_0, \tilde{x}_0)} + \log \frac{p(e)p(\tilde{e}|e, I)}{q(e, \tilde{e}|x_0, \tilde{x}_0, I)} + \log \frac{p(x_0, u|e)}{q(u|x_0)} + \log \frac{p(\tilde{x}_0, \tilde{u}|\tilde{e})}{q(\tilde{u}|\tilde{x}_0)}\right] \\
&=\mathbb{E}_{q(I|x_0,\tilde{x}_0)}\mathbb{E}_{q(e,\tilde{e}|x_0,\tilde{x}_0,I)}\mathbb{E}_{q(u|x_0)}\mathbb{E}_{q(\tilde{u}|\tilde{x}_0)}\Bigg[\Big[\log p(I) + \log p(e) + \log p(\tilde{e}|e, I) \\
&- \log q(I|x_0, \tilde{x}_0) - \log q(e, \tilde{e}|x_0, \tilde{x}_0, I)\Big] + \left[\log \frac{p(x_0, u|e)}{q(u|x_0)} + \log \frac{p(\tilde{x}_0, \tilde{u}|\tilde{e})}{q(\tilde{u}|\tilde{x}_0)}\right]\Bigg]
\end{aligned}
$$

The terms in the first bracket correspond to the intervention encoder and the noise encoding module, respectively, and the terms in the second bracket correspond to the diffusion model conditioned on pre- and post-intervention noise encodings.

Song et al. [53] shows that the discretization of SDE formulations of the diffusion model is equivalent to discrete-time diffusion models. Therefore, for simplicity, we derive the ELBO for discrete-time diffusion models. Following [73], for a discrete-time diffusion model where $t \in [1, T]$, we have

$$\begin{aligned}\mathbb{E}_{q(I|x_0,\tilde{x}_0)}\mathbb{E}_{q(e,\tilde{e}|x_0,\tilde{x}_0,I)}\mathbb{E}_{q(u|x_0)}\mathbb{E}_{q(\tilde{u}|\tilde{x}_0)}\left[\log\frac{p(x_0,u|e)}{q(u|x_0)}\right] &= \\ \mathbb{E}_{q(e,\tilde{e}|x_0,\tilde{x}_0,I)}\Bigg[\mathbb{E}_{q(u_1|x_0)}[\log p(x_0|u_1)] - D_{KL}(q(u_T|x_0)||p(u_T)) \\ - \sum_{t=2}^{T}\mathbb{E}_{q(u_t|x_0)}[D_{KL}(q(u_{t-1}|u_t,x_0,e)||p(u_{t-1}|u_t,e)]\Bigg]\end{aligned} \tag{A1}$$

where we have the following:

- $\mathbb{E}_{q(u_1|x_0)}[\log p(x_0|u_1)]$ is the reconstruction term, and it can be defined in a way that it is constant so it can be ignored during training.
- $D_{KL}(q(u_T|x_0)||p(u_T))$ is the prior matching term and can similarly be defined in a way that it is constant.
- $\mathbb{E}_{u_t|x_0}[D_{KL}(q(u_{t-1}|u_t,x_0,e)||p(u_{t-1}|u_t,e)]$ is a denoising matching term. This term is the origin of different interpretations of the score-based diffusion models.

For the SDE formulation of the forward diffusion process, the denoising matching term becomes [53]

$$\lambda(t)||s_\theta(u_t,e,t) - \nabla_{u_t}\log p(u_t|x_0)||_2^2. \tag{A2}$$

The weight $\lambda(t)$ of denoising matching terms is related to the diffusion coefficient of the forward SDE. For a Variance Exploding SDE, the weight is defined as $\lambda(t) = 2\sigma^2(t)\log(\sigma_{max}/\sigma_{min})$ with $\sigma(t) = \sigma_{min}\cdot(\sigma_{max}/\sigma_{min})^t$. Therefore, by combining (A1) with (A2), the ELBO becomes

$$\begin{aligned}\log p(x_0,\tilde{x}_0) \geq\ & \mathbb{E}_{p(x_0,\tilde{x}_0)}\mathbb{E}_{q(I|x_0,\tilde{x}_0)}\mathbb{E}_{q(e,\tilde{e}|x_0,\tilde{x}_0,I)}\mathbb{E}_{t\sim U(0,1)}\mathbb{E}_{q(u_t|x_0)}\mathbb{E}_{q(\tilde{u}_t|\tilde{x}_0)} \\ & \Bigg[\log p(I) + \log p(e) + \log p(\tilde{e}|e,I) - \log q(I|x_0,\tilde{x}_0) - \log q(e,\tilde{e}|x_0,\tilde{x}_0,I) \\ & +\lambda(t)\left[||s_\theta(u_t,e,t) - \nabla_{u_t}\log p(u_t|x_0)||_2^2 + ||s_\theta(\tilde{u}_t,\tilde{e},t) - \nabla_{\tilde{u}_t}\log p(\tilde{u}_t|\tilde{x}_0)||_2^2\right]\Bigg]\end{aligned}$$

For infinite-dimensional representations, we can derive the ELBO using a similar argument. In this case, the formula for the ELBO is

$$\begin{aligned}\log p(x_0,\tilde{x}_0) \geq\ & \mathbb{E}_{p(x_0,\tilde{x}_0)}\mathbb{E}_{q(I|x_0,\tilde{x}_0)}\mathbb{E}_{t\sim U(0,1)}\mathbb{E}_{q(e_t,\tilde{e}_t|x_0,\tilde{x}_0,I)}\mathbb{E}_{q(u_t|x_0)}\mathbb{E}_{q(\tilde{u}_t|\tilde{x}_0)} \\ & \Bigg[\log p(I) + \log p(e_t) + \log p(\tilde{e}_t|e_t,I) - \log q(I|x_0,\tilde{x}_0) - \log q(e_t,\tilde{e}_t|x_0,\tilde{x}_0,I) \\ & +\lambda(t)||s_\theta(u_t,e_t,t) - \nabla_{u_t}\log p(u_t|x_0)||_2^2 + \lambda(t)||s_\theta(\tilde{u}_t,\tilde{e}_t,t) - \nabla_{\tilde{u}_t}\log p(\tilde{u}_t|\tilde{x}_0)||_2^2\Bigg]\end{aligned}$$

Appendix C. Implementation Details

Appendix C.1. Training

For the training, we follow the four-phase training of Brehmer et al. [14] but consider only the first three phases. In summary, we consider the following steps:

(1) We begin by training the diffusion model and the encoding module together on data pairs for 20 epochs. This can be interpreted as a warm-up for the diffusion model and the encoding module to extract meaningful representations of data.

(2) In the second phase, we include all modules for training, except for solution functions. We consider $p(\tilde{e}_i|e)$ to be a uniform probability density. The framework is trained in this phase for 50 epochs.

(3) We include solution functions and train the whole framework with the proposed loss for 50 epochs.

We find out that considering our data generation process, including the fourth training phase of Brehmer et al. [14] has no impact on the model's performance. Consequently, we choose to disregard it in our analysis. We use the loss in Equation (3) as the objective function and consider the coefficient of the regularization term $\mathcal{L}_{entropy}$ to be 1. Therefore, our overall loss function is then given by $\mathcal{L} = \mathcal{L}_{model} + \mathcal{L}_{entropy}$.

Appendix C.2. Architectures and Hyperparameters

We train the model for 120 epochs and use the learning rate of 3e-4 with a batch size of 64. β is initially set to 0 and increased to 1 during training. The noise encoder is considered Gaussian, with the mean and standard deviation parameterized as an MLP with two hidden layers and 64 units each and ReLU activation functions. The solution functions are implemented as affine transformations, where the slope and offset are learned from pre-intervention noise encodings. These functions utilize the same architecture as the noise encoder for learning the slope and offset parameters. The architecture of the score function of the diffusion model is based on NCSN++ architecture [53] with the same set of hyperparameters used for the CIFAR-10 dataset. As the input x is 16-dimensional and the score model follows a convolutional architecture, we reshape the input into a 4×4 format and then feed it into the diffusion model. Furthermore, in the forward SDE, σ_{min} and σ_{max} are set to 0.01 and 50, respectively.

References

1. Schölkopf, B.; Locatello, F.; Bauer, S.; Ke, N.R.; Kalchbrenner, N.; Goyal, A.; Bengio, Y. Toward causal representation learning. *Proc. IEEE* **2021**, *109*, 612–634. [CrossRef]
2. Hellström, T. The relevance of causation in robotics: A review, categorization, and analysis. *Paladyn J. Behav. Robot.* **2021**, *12*, 238–255. [CrossRef]
3. Anwar, A.R.; Mideska, K.G.; Hellriegel, H.; Hoogenboom, N.; Krause, H.; Schnitzler, A.; Deuschl, G.; Raethjen, J.; Heute, U.; Muthuraman, M. Multi-modal causality analysis of eyes-open and eyes-closed data from simultaneously recorded EEG and MEG. In Proceedings of the 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, IL, USA, 26–30 August 2014; pp. 2825–2828.
4. Runge, J.; Bathiany, S.; Bollt, E.; Camps-Valls, G.; Coumou, D.; Deyle, E.; Glymour, C.; Kretschmer, M.; Mahecha, M.D.; Muñoz-Marí, J.; et al. Inferring causation from time series in Earth system sciences. *Nat. Commun.* **2019**, *10*, 2553. [CrossRef]
5. Hernán, M.Á.; Brumback, B.; Robins, J.M. Marginal structural models to estimate the causal effect of zidovudine on the survival of HIV-positive men. *Epidemiology* **2000**, *11*, 561–570. [CrossRef]
6. Robins, J.M.; Hernan, M.A.; Brumback, B. Marginal structural models and causal inference in epidemiology. *Epidemiology* **2000**, *11*, 550–560. [CrossRef] [PubMed]
7. Hiemstra, C.; Jones, J.D. Testing for linear and nonlinear Granger causality in the stock price-volume relation. *J. Financ.* **1994**, *49*, 1639–1664.
8. Kıcıman, E.; Ness, R.; Sharma, A.; Tan, C. Causal reasoning and large language models: Opening a new frontier for causality. *arXiv* **2023**, arXiv:2305.00050.
9. Lampinen, A.; Chan, S.; Dasgupta, I.; Nam, A.; Wang, J. Passive learning of active causal strategies in agents and language models. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
10. Zečević, M.; Willig, M.; Dhami, D.S.; Kersting, K. Causal parrots: Large language models may talk causality but are not causal. *arXiv* **2023**, arXiv:2308.13067.
11. Yang, M.; Liu, F.; Chen, Z.; Shen, X.; Hao, J.; Wang, J. Causalvae: Structured causal disentanglement in variational autoencoder. *arXiv* **2020**, arXiv:2004.08697.
12. Liu, Y.; Zhang, Z.; Gong, D.; Gong, M.; Huang, B.; Hengel, A.v.d.; Zhang, K.; Shi, J.Q. Identifying Weight-Variant Latent Causal Models. *arXiv* **2022**, arXiv:2208.14153.
13. Subramanian, J.; Annadani, Y.; Sheth, I.; Ke, N.R.; Deleu, T.; Bauer, S.; Nowrouzezahrai, D.; Kahou, S.E. Learning Latent Structural Causal Models. *arXiv* **2022**, arXiv:2210.13583.
14. Brehmer, J.; De Haan, P.; Lippe, P.; Cohen, T.S. Weakly supervised causal representation learning. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 38319–38331.

15. Ahuja, K.; Mahajan, D.; Wang, Y.; Bengio, Y. Interventional causal representation learning. In Proceedings of the International Conference on Machine Learning, PMLR, Honolulu, HI, USA, 23–29 July 2023; pp. 372–407.
16. Zhang, J.; Greenewald, K.; Squires, C.; Srivastava, A.; Shanmugam, K.; Uhler, C. Identifiability guarantees for causal disentanglement from soft interventions. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
17. Jiang, Y.; Aragam, B. Learning nonparametric latent causal graphs with unknown interventions. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
18. Zhang, J.; Spirtes, P. Strong faithfulness and uniform consistency in causal inference. In Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence, Edmonton, AB, Canada, 1–4 August 2002; pp. 632–639.
19. Van Steenkiste, S.; Locatello, F.; Schmidhuber, J.; Bachem, O. Are disentangled representations helpful for abstract visual reasoning? *Adv. Neural Inf. Process. Syst.* **2019**, *32*.
20. Dittadi, A.; Papa, S.; De Vita, M.; Schölkopf, B.; Winther, O.; Locatello, F. Generalization and Robustness Implications in Object-Centric Learning. *arXiv* **2021**, arXiv:2107.00637.
21. Wu, Z.; Dvornik, N.; Greff, K.; Kipf, T.; Garg, A. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *arXiv* **2022**, arXiv:2210.05861.
22. Yoon, J.; Wu, Y.F.; Bae, H.; Ahn, S. An investigation into pre-training object-centric representations for reinforcement learning. *arXiv* **2023**, arXiv:2302.04419.
23. Papa, S.; Winther, O.; Dittadi, A. Inductive Biases for Object-Centric Representations in the Presence of Complex Textures. In Proceedings of the UAI 2022 Workshop on Causal Representation Learning, Eindhoven, The Netherlands, 5–6 August 2022.
24. Mansouri, A.; Hartford, J.; Zhang, Y.; Bengio, Y. Object-centric architectures enable efficient causal representation learning. *arXiv* **2023**, arXiv:2310.19054.
25. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
26. Locatello, F.; Poole, B.; Rätsch, G.; Schölkopf, B.; Bachem, O.; Tschannen, M. Weakly-supervised disentanglement without compromises. In Proceedings of the International Conference on Machine Learning, PMLR, Virtual, 13–18 July 2020; pp. 6348–6359.
27. Dhariwal, P.; Nichol, A. Diffusion models beat gans on image synthesis. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 8780–8794.
28. Ramesh, A.; Dhariwal, P.; Nichol, A.; Chu, C.; Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. *arXiv* **2022**, arXiv:2204.06125.
29. Ho, J.; Saharia, C.; Chan, W.; Fleet, D.J.; Norouzi, M.; Salimans, T. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.* **2022**, *23*, 1–33.
30. Saharia, C.; Chan, W.; Saxena, S.; Li, L.; Whang, J.; Denton, E.L.; Ghasemipour, K.; Gontijo Lopes, R.; Karagol Ayan, B.; Salimans, T.; et al. Photorealistic text-to-image diffusion models with deep language understanding. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 36479–36494.
31. Ho, J.; Chan, W.; Saharia, C.; Whang, J.; Gao, R.; Gritsenko, A.; Kingma, D.P.; Poole, B.; Norouzi, M.; Fleet, D.J.; et al. Imagen video: High definition video generation with diffusion models. *arXiv* **2022**, arXiv:2210.02303.
32. Ho, J.; Salimans, T.; Gritsenko, A.; Chan, W.; Norouzi, M.; Fleet, D.J. Video Diffusion Models. *arXiv* **2022**, arXiv:2204.03458.
33. Hatamizadeh, A.; Song, J.; Liu, G.; Kautz, J.; Vahdat, A. Diffit: Diffusion vision transformers for image generation. *arXiv* **2023**, arXiv:2312.02139.
34. Kim, D.; Kim, Y.; Kwon, S.J.; Kang, W.; Moon, I.C. Refining generative process with discriminator guidance in score-based diffusion models. *arXiv* **2022**, arXiv:2211.17091.
35. Kong, Z.; Ping, W.; Huang, J.; Zhao, K.; Catanzaro, B. Diffwave: A versatile diffusion model for audio synthesis. *arXiv* **2020**, arXiv:2009.09761.
36. Huang, Q.; Park, D.S.; Wang, T.; Denk, T.I.; Ly, A.; Chen, N.; Zhang, Z.; Zhang, Z.; Yu, J.; Frank, C.; et al. Noise2music: Text-conditioned music generation with diffusion models. *arXiv* **2023**, arXiv:2302.03917.
37. Ruan, L.; Ma, Y.; Yang, H.; He, H.; Liu, B.; Fu, J.; Yuan, N.J.; Jin, Q.; Guo, B. Mm-diffusion: Learning multi-modal diffusion models for joint audio and video generation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 10219–10228.
38. Watson, J.L.; Juergens, D.; Bennett, N.R.; Trippe, B.L.; Yim, J.; Eisenach, H.E.; Ahern, W.; Borst, A.J.; Ragotte, R.J.; Milles, L.F.; et al. De novo design of protein structure and function with RFdiffusion. *Nature* **2023**, *620*, 1089–1100. [CrossRef] [PubMed]
39. Wu, K.E.; Yang, K.K.; van den Berg, R.; Alamdari, S.; Zou, J.Y.; Lu, A.X.; Amini, A.P. Protein structure generation via folding diffusion. *Nat. Commun.* **2024**, *15*, 1059. [CrossRef] [PubMed]
40. Gruver, N.; Stanton, S.; Frey, N.; Rudner, T.G.; Hotzel, I.; Lafrance-Vanasse, J.; Rajpal, A.; Cho, K.; Wilson, A.G. Protein design with guided discrete diffusion. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
41. Luo, S.; Su, Y.; Peng, X.; Wang, S.; Peng, J.; Ma, J. Antigen-specific antibody design and optimization with diffusion-based generative models for protein structures. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 9754–9767.
42. Chen, X.; Liu, Z.; Xie, S.; He, K. Deconstructing denoising diffusion models for self-supervised learning. *arXiv* **2024**, arXiv:2401.14404.
43. Abstreiter, K.; Mittal, S.; Bauer, S.; Schölkopf, B.; Mehrjou, A. Diffusion-Based Representation Learning. *arXiv* **2022**, arXiv:2105.14257.

44. Mittal, S.; Lajoie, G.; Bauer, S.; Mehrjou, A. From Points to Functions: Infinite-dimensional Representations in Diffusion Models. *arXiv* 2022, arXiv:2210.13774.
45. Wang, Y.; Schiff, Y.; Gokaslan, A.; Pan, W.; Wang, F.; De Sa, C.; Kuleshov, V. InfoDiffusion: Representation Learning Using Information Maximizing Diffusion Models. *arXiv* **2023**, arXiv:2306.08757.
46. Kwon, M.; Jeong, J.; Uh, Y. Diffusion models already have a semantic latent space. *arXiv* **2022**, arXiv:2210.10960.
47. Zhang, Z.; Zhao, Z.; Lin, Z. Unsupervised representation learning from pre-trained diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 22117–22130.
48. Traub, J. Representation Learning with Diffusion Models. *arXiv* **2022**, arXiv:2210.11058.
49. Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the International Conference on Machine Learning, PMLR, Lille, France, 6–11 July 2015; pp. 2256–2265.
50. Niu, C.; Song, Y.; Song, J.; Zhao, S.; Grover, A.; Ermon, S. Permutation invariant graph generation via score-based generative modeling. In Proceedings of the International Conference on Artificial Intelligence and Statistics, PMLR, Virtual, 26–28 August 2020; pp. 4474–4484.
51. Ho, J.; Jain, A.; Abbeel, P. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 6840–6851.
52. Song, J.; Meng, C.; Ermon, S. Denoising diffusion implicit models. *arXiv* **2020**, arXiv:2010.02502.
53. Song, Y.; Sohl-Dickstein, J.; Kingma, D.P.; Kumar, A.; Ermon, S.; Poole, B. Score-Based Generative Modeling through Stochastic Differential Equations. *arXiv* **2021**, arXiv:2011.13456.
54. Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; Ommer, B. High-resolution image synthesis with latent diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10684–10695.
55. Kocaoglu, M.; Snyder, C.; Dimakis, A.G.; Vishwanath, S. ausalGAN: Learning Causal Implicit Generative Models with Adversarial Training. *arXiv* **2017**, arXiv:1709.02023.
56. Komanduri, A.; Wu, Y.; Huang, W.; Chen, F.; Wu, X. SCM-VAE: Learning Identifiable Causal Representations via Structural Knowledge. In Proceedings of the 2022 IEEE International Conference on Big Data, Osaka, Japan, 17–20 December 2022; pp. 1014–1023.
57. Von Kügelgen, J.; Sharma, Y.; Gresele, L.; Brendel, W.; Schölkopf, B.; Besserve, M.; Locatello, F. Self-supervised learning with data augmentations provably isolates content from style. *Proc. NeurIPS* **2021**, *34*, 16451–16467.
58. Sturma, N.; Squires, C.; Drton, M.; Uhler, C. Unpaired multi-domain causal representation learning. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
59. Buchholz, S.; Rajendran, G.; Rosenfeld, E.; Aragam, B.; Schölkopf, B.; Ravikumar, P. Learning linear causal representations from interventions under general nonlinear mixing. *Adv. Neural Inf. Process. Syst.* **2024**, *36*.
60. Sanchez, P.; Tsaftaris, S.A. Diffusion causal models for counterfactual estimation. *arXiv* **2022**, arXiv:2202.10166.
61. Sanchez, P.; Liu, X.; O'Neil, A.Q.; Tsaftaris, S.A. Diffusion models for causal discovery via topological ordering. *arXiv* **2022**, arXiv:2210.06201.
62. Locatello, F.; Bauer, S.; Lucic, M.; Raetsch, G.; Gelly, S.; Schölkopf, B.; Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 4114–4124.
63. Shu, R.; Chen, Y.; Kumar, A.; Ermon, S.; Poole, B. Weakly supervised disentanglement with guarantees. *arXiv* **2019**, arXiv:1910.09772.
64. Lachapelle, S.; Rodriguez, P.; Sharma, Y.; Everett, K.E.; Le Priol, R.; Lacoste, A.; Lacoste-Julien, S. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In Proceedings of the Conference on Causal Learning and Reasoning, Eureka, CA, USA, 11–13 April 2022; pp. 428–484.
65. Hyvärinen, A.; Oja, E. Independent component analysis: Algorithms and applications. *Neural Netw.* **2000**, *13*, 411–430. [CrossRef]
66. Khemakhem, I.; Kingma, D.; Monti, R.; Hyvarinen, A. Variational autoencoders and nonlinear ica: A unifying framework. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Virtual, 26–28 August 2020; pp. 2207–2217.
67. Pearl, J. *Causality*; Cambridge University Press: Cambridge, UK, 2009.
68. Bongers, S.; Forré, P.; Peters, J.; Mooij, J.M. Foundations of structural causal models with cycles and latent variables. *Ann. Stat.* **2021**, *49*, 2885–2915. [CrossRef]
69. Hyvärinen, A.; Dayan, P. Estimation of non-normalized statistical models by score matching. *J. Mach. Learn. Res.* **2005**, *6*, 695–709.
70. Vincent, P. A connection between score matching and denoising autoencoders. *Neural Comput.* **2011**, *23*, 1661–1674. [CrossRef] [PubMed]
71. Lippe, P.; Cohen, T.; Gavves, E. Efficient neural causal discovery without acyclicity constraints. *arXiv* **2021**, arXiv:2107.10483.
72. Eastwood, C.; Williams, C.K. A framework for the quantitative evaluation of disentangled representations. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
73. Luo, C. Understanding diffusion models: A unified perspective. *arXiv* **2022**, arXiv:2208.11970.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.