arXiv:2010.02637v1 [cs.LG] 6 Oct 2020

Disentangled Generative Causal Representation Learning

Xinwei Shen*, Furui Liu†, Hanze Dong*, Qing Lian*, Zhitang Chen†, and Tong Zhang*

* The Hong Kong University of Science and Technology
† Huawei Noah's Ark Lab

Abstract

This paper proposes a Disentangled gEnerative cAusal Representation (DEAR) learning method. Unlike existing disentanglement methods that enforce independence of the latent variables, we consider the general case where the underlying factors of interests can be causally correlated. We show that previous methods with independent priors fail to disentangle causally correlated factors. Motivated by this finding, we propose a new disentangled learning method called DEAR that enables causal controllable generation and causal representation learning. The key ingredient of this new formulation is to use a structural causal model (SCM) as the prior for a bidirectional generative model. The prior is then trained jointly with a generator and an encoder using a suitable GAN loss. Theoretical justification on the proposed formulation is provided, which guarantees disentangled causal representation learning under appropriate conditions. We conduct extensive experiments on both synthesized and real datasets to demonstrate the effectiveness of DEAR in causal controllable generation, and the benefits of the learned representations for downstream tasks in terms of sample efficiency and distributional robustness.

1 Introduction

Consider the observed data x from a distribution q_x on $\mathcal{X} \subseteq \mathbb{R}^d$ and the latent variable z from a prior p_z on $\mathcal{Z} \subseteq \mathbb{R}^k$. In bidirectional generative models (BGMs), we are normally interested in learning an *encoder* $E : \mathcal{X} \to \mathcal{Z}$ to infer latent variables and a *generator* $G : \mathcal{Z} \to \mathcal{X}$ to generate data, to achieve both representation learning and data generation. Classical BGMs include Variational Autoencoder (VAE) (Kingma & Welling, 2014) and BiGAN (Donahue et al., 2017). In representation learning, it was argued that an effective representation for downstream learning tasks should disentangle the underlying factors of variation (Bengio et al., 2013). In generation, it is highly desirable if one can control the semantic generative factors by aligning them with the latent variables such as in StyleGAN (Karras et al., 2019). Both goals can be achieved with the disentanglement of latent variable z, which informally means that each dimension of z measures a distinct factor of variation in the data (Bengio et al., 2013).

Earlier unsupervised disentanglement methods mostly regularize the VAE objective to encourage independence of learned representations (Higgins et al., 2017; Burgess et al., 2017; Kim & Mnih, 2018; Chen et al., 2018; Kumar et al., 2018). Later, Locatello et al. (2019) show that unsupervised

learning of disentangled representations is impossible: many existing unsupervised methods are actually brittle, requiring careful supervised hyperparameter tuning or implicit inductive biases. To promote identifiability, recent work resorts to various forms of supervision (Locatello et al., 2020b; Shu et al., 2020; Locatello et al., 2020a). In this work, we also incorporate supervision on the ground-truth factors in the form stated in Section 3.2.

Most of these existing methods are built on the assumption that the underlying factors of variation are mutually independent. However, in many real world cases the semantically meaningful factors of interests are not independent (Bengio et al., 2020). Instead, semantically meaningful high-level variables are often causally correlated, *i.e.*, connected by a causal graph. In this paper, we prove formally that methods with independent priors fail to disentangle causally correlated factors. Motivated by this observation, we propose a new method to learn disentangled generative causal representations called DEAR. The key ingredient of our formulation is a structured causal model (SCM) (Pearl et al., 2000) as the prior for latent variables in a bidirectional generative model. The causal prior is then learned jointly with a generator and an encoder using a suitable GAN (Goodfellow et al., 2014) loss. We establish theoretical guarantees for DEAR to learn disentangled causal representations under appropriate conditions.

An immediate application of DEAR is causal controllable generation, which can generate data from any desired interventional distributions of the latent factors. Another useful application of disentangled representations is to use such representations in downstream tasks, leading to better sample complexity (Bengio et al., 2013; Schölkopf et al., 2012). Moreover, it is believed that causal disentanglement is invariant and thus robust under distribution shifts (Schölkopf, 2019; Arjovsky et al., 2019). In this paper, we demonstrate these conjectures in various downstream prediction tasks for the proposed DEAR method, which has theoretically guaranteed disentanglement property.

We summarize our main contributions as follows:

- We formally identify a problem with previous disentangled representation learning methods using the independent prior assumption, and prove that they fail to disentangle when the underlying factors of interests are causally correlated.
- We propose a new disentangled learning method, DEAR, which integrates an SCM prior into a bidirectional generative model, trained with a suitable GAN loss.
- We provide theoretical justification on the identifiability of the proposed formulation.
- Extensive experiments are conducted on both synthesized and real data to demonstrate the effectiveness of DEAR in causal controllable generation, and the benefits of the learned representations for downstream tasks in terms of sample efficiency and distributional robustness.

2 Other related work

GAN-based disentanglement methods. Existing methods, including InfoGAN (Chen et al., 2016) and InfoGAN-CR (Lin et al., 2020), differ from our proposed formulation mainly in two folds. First they still assume an independent prior for latent variables, so suffer from the same problem with previous VAE-based methods mentioned above. Besides, the idea of InfoGAN-CR is to encourage each latent code to make changes that are easy to detect, which actually applies well

only when the underlying factors are independent. Second, InfoGAN as a bidirectional generative modeling method further requires variational approximation apart from adversarial training, which is inferior to the principled formulation in BiGAN and AGES (Shen et al., 2020) that we adopt.

Causality with generative models. CausalGAN (Kocaoglu et al., 2018) and a concurrent work (Moraffah et al., 2020) of ours, are unidirectional generation models that build upon a cGAN (Mirza & Osindero, 2014) and assign an SCM to the conditional attributes while leave the latent variables as independent Gaussian noises. The limit of a cGAN is that it always requires full supervision on attributes to apply conditional adversarial training. Moreover their unidirectional nature makes it impossible to learn representations, and they have nothing to do with disentanglement learning due to the direct access to the attributes. Besides they only consider binary factors whose consequent semantic interpolations appear non-smooth, as shown in Appendix D. CausalVAE (Yang et al., 2020) assigns the SCM directly on the latent variables, but built upon iVAE (Khemakhem et al., 2020), it adopts a conditional prior given the ground-truth factors so is also limited to fully supervised setting.

Structured latent space in a generative model. VLAE (Zhao et al., 2017) and SAE (Leeb et al., 2020) decompose the latent space into separate chunks each of which is processed at different levels of the encoder and decoder (generator). VQ-VAE-2 (Razavi et al., 2019) uses a two-level latent space along with a multi-stage generation mechanism to capture both high and low level information of data. These methods essentially adopt implicit probabilistic or architectural hierarchies, in contrast to the causal structure that we impose to the latent space, and thus cannot achieve the goal of causal disentanglement.

3 Problem setting

3.1 Generative model

We first describe the probabilistic framework of disentangled learning with supervision. Note that the encoder and generator are generally stochastic. During the inference process, the encoder induces the encoded conditional $q_E(z|x)$ which can be a factorized Gaussian and the encoded joint distribution $q_E(x,z) = q_x(x)q_E(z|x)$. During the generation process, the generator induces the generated conditional $p_G(x|z)$ and generated joint distribution $p_G(x,z) = p_z(z)p_G(x|z)$. We consider the following objective for generative modeling:

$$L_{\text{gen}}(E,G) = D_{\text{KL}}(q_E(x,z), p_G(x,z)), \tag{1}$$

where D_{KL} is the Kullback-Leibler (KL) divergence. (1) is shown to be equivalent to the evidence lower bound used in VAEs up to a constant, and allows a closed form only with factorized Gaussian prior, encoder and generator (Shen et al., 2020). Since constraints are required to enforce disentanglement of the latent space, it is desired that the distribution family of $q_E(x,z)$ and $p_G(x,z)$ should be large enough, especially for complex data like images. Normally more general implicit distributions are favored over factorized Gaussians in terms of expressiveness (Karras et al., 2019; Mescheder et al., 2017). Then minimizing (1) requires adversarial training, as discussed detailedly in Section 4.3.

3.2 Supervised regularizer

To guarantee disentanglement, we incorporate supervision when training the BGM, following the similar idea in Locatello et al. (2020b) but with a different formulation. Specifically, let $\xi \in \mathbb{R}^m$ be the underlying factors of x, and y_i be some continuous or discrete observation of the underlying factor ξ_i satisfying $\xi_i = \mathbb{E}(y_i|x)$ for $i = 1, \dots, m$.

Let $\bar{E}(x)$ be the deterministic part of the stochastic transformation $E(x)$, i.e., $\bar{E}(x) = \mathbb{E}(E(x)|x)$, which is used for representation learning. We consider the following objective:

$$L(E, G) = L_{\text{gen}}(E, G) + \lambda L_{\text{sup}}(E), \tag{2}$$

where $L_{\text{sup}} = \sum_{i=1}^{m} \mathbb{E}_{(x,y)}[\text{CE}(\bar{E}_i(x), y_i)]$ if y_i is the binary or bounded (and normalized to $[0, 1]$) continuous label of the i-th factor ξ_i, where $\text{CE}(l, y) = y \log \sigma(l) + (1 - y) \log(1 - \sigma(l))$ is the cross-entropy loss with $\sigma(\cdot)$ being the sigmoid function; $L_{\text{sup}}(\phi) = \sum_{i=1}^{m} \mathbb{E}_{(x,y)}[\bar{E}_i(x) - y_i]^2$ if y_i is the continuous observation of ξ_i, and $\lambda > 0$. Estimating of L_{gen} requires the unlabelled dataset $\{x^1, \dots, x^N\}$, while estimating L_{sup} requires a labeled dataset $\{(x^j, y^j) : j = 1, \dots, N_s\}$ where N_s can be much smaller than N.

3.3 Unidentifiability with an independent prior

Intuitively, the above supervised regularizer aims at ensuring some alignment between factor ξ and latent variable z. We start with the definition of a disentangled representation following this intuition.

Definition 1 (Disentangled representation)**.** *Given the underlying factor $\xi \in \mathbb{R}^m$ of data x, a deterministic encoder E is said to learn a disentangled representation with respect to ξ if $\forall i = 1, \dots, m$, there exists a 1-1 function g_i such that $E_i(x) = g_i(\xi_i)$. Further, a stochastic encoder E is said to be disentangled wrt ξ if its deterministic part $\bar{E}(x)$ is disentangled wrt ξ.*

As stated above, we consider the general case where the underlying factors of interests are causally correlated, meaning that the elements of ξ are connected by a causal graph whose adjacency matrix is not a diagonal matrix. Then the goal becomes to disentangle the causal factors. Previous methods mostly use an independent prior for z, which contradicts with the truth. We make this formal through the following proposition, which indicates that the disentangled representation is generally unidentifiable with an independent prior.

Proposition 1. *Let E^* be any encoder that is disentangled wrt ξ. Let $b^* = L_{\text{sup}}(E^*)$, $a = \min_G L_{\text{gen}}(E^*, G)$, and $b = \min_{\{(E,G):L_{\text{gen}}=0\}} L_{\text{sup}}(E)$. Suppose the prior p_z is factorized, i.e., $p_z(z) = \prod_{i=1}^{k} p_i(z_i)$. Then we have $a > 0$, and either when $b^* \geq b$ or $b^* < b$ and $\lambda < \frac{a}{b-b^*}$, there exists a solution (E', G') such that for any generator G, we have $L(E', G') < L(E^*, G)$.*

This proposition directly suggests that minimizing (1) favors the solution (E', G') over one with a disentangled encoder E^*. Thus, with an independent prior we have no way to identify the disentangled solution with λ that is not large enough. However, in real applications it is impossible to estimate the threshold, and too large λ makes it difficult to learn the BGM. In the following section we propose a solution to this problem.

4 Causal disentanglement learning

4.1 Generative model with a causal prior

We propose to use a causal model as the prior p_z. Specifically we use the general nonlinear Structural Causal Model (SCM) (Yu et al., 2019) as follows

$$z = f((I - A^\top)^{-1} h(\epsilon)) := F_\beta(\epsilon), \tag{3}$$

where $A \in \mathbb{R}^{k \times k}$ is the weighted adjacency matrix of the directed acyclic graph (DAG) upon the k elements of z (*i.e.*, $A_{ij} \neq 0$ if and only if z_i is the parent of z_j), f and h are element-wise nonlinear transformations, and β is the set of parameters of f, h and A. When f is invertible, (3) is equivalent to

$$f^{-1}(z) = A^\top f^{-1}(z) + h(\epsilon) \tag{4}$$

which indicates that the factors z satisfy a linear SCM after nonlinear transformation f, and enables interventions on latent variables as discussed later. The model structure is presented in Figure 1.

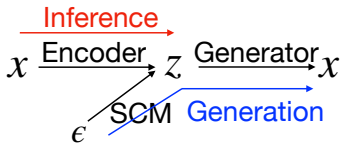


Figure 1: Model structure of a BGM with an SCM prior.

In causal structure learning, the graph is required to be acyclic. Zheng et al. (2018) propose an equality constraint whose satisfaction ensures acyclicity and solve the problem with augmented Lagrangian method, which however leads to optimization difficulties (Ng et al., 2020). In this paper, to avoid dealing with the non-convex constraint but focus on disentangling, we assume to have some prior knowledge of the binary causal structure. Specifically, we assume the super-graph of the true graph is given, the best case of which is the true graph while the worst is that only the causal order is available. Then we learn the weights of the non-zero elements of the prior adjacency matrix that indicate the direction and scale of causal effects, jointly with other parameters using the formulation and algorithm described in later sections. To incorporate structure learning methods and jointly learn the structure from scratch with guarantee of identifiability could be explored in future work.

To enable causal controllable generation, we use invertible f and h and describe the mechanism to generate images from interventional distributions of latent variables. Suppose we would like to intervene on the i-th dimension of z, i.e., $\mathrm{Do}(z_i = c)$, where c is a constant. Once we have latent factors z inferred from data x, i.e., $z = E(x)$, or sampled from prior p_z, we first obtain the corresponding noise $\epsilon = F^{-1}(z)$, and then follow the intervened equations in (4) to obtain z' on the left hand side using ancestral sampling by performing (4) iteratively. Then we decode the intervened latent factor z' to generate the sample $G(z')$.

Another issue of the model is the latent dimension, to handle which we propose the so-called composite prior. Suppose there are m generative factors that we are interested to disentangle, e.g.,

all the semantic concepts related to some field, where m tends to be smaller than the total number of generative factors. If we set the latent dimension $k = m$, the BGM will lose enough capacity to generate or reconstruct the data. Hence we propose to set $k > m$ and use a prior that is a composition of a causal model for the first m dimensions and another distribution for the other $k - m$ dimensions, like a standard Gaussian.

4.2 Formulation and identifiability of disentanglement

In this section, we present the formulation of DEAR and establish the theoretical justification on it. Compared with the BGM described in Section 3.1, here we have one more module to learn that is the SCM prior. Thus $p_G(x, z)$ becomes $p_{G,F}(x, z) = p_F(z)p_G(x|z)$ where $p_F(z)$ is the marginal distribution of $F_\beta(\epsilon)$ with $\epsilon \sim \mathcal{N}(0, I)$. We then rewrite the generative loss as follows

$$L_{\text{gen}}(E, G, F) = D_{\text{KL}}(q_E(x, z), p_{G,F}(x, z)). \tag{5}$$

Then we propose the following formulation to learn causal generative causal representations:

$$\min_{E,G,F} \; L(E, G, F) = L_{\text{gen}}(E, G, F) + \lambda L_{\text{sup}}(E). \tag{6}$$

The following theorem guarantees that the DEAR formulation can learn disentangled representations defined in Definition 1 when the underlying factors are causally correlated.

Theorem 1. *Assume the infinite capacity of E, G and f. Further assume the true binary adjacency matrix can be learned.*[1] *Then DEAR learns the disentangled encoder E^*. Specifically, we have $g_i(\xi_i) = \sigma^{-1}(\xi_i)$ if CE loss is used in the supervised regularizer, and $g_i(\xi_i) = \xi_i$ if L_2 loss is used.*

Note that the identifiability we establish in this paper differs from some previous work on the parameter identifiability, e.g., Khemakhem et al. (2020). We argue that to learn disentangled representations, the form in Definition 1, i.e., the existence but not the uniqueness of g_i's, is sufficient to identify the relation among the representations and the data. In contrast, parameter identifiability may not be achievable in many cases like over-parametrization. Thus the identifiability discussed here is more realistic in terms of the goal of disentangling. Later we provide empirical evidence to support the theory directly through the application in causal controllable generation.

4.3 Algorithm

In this section we propose the algorithm to solve the above formulation (6). The SCM prior $p_F(z)$ and implicit generated conditional $p_G(x|z)$ makes (5) lose an analytic form. Hence we adopt a GAN method to adversarially estimate the gradient of (5) as in Shen et al. (2020). We parametrize $E_\phi(x)$ and $G_\theta(z)$ by neural networks. Different from previous work, the prior also involves learnable parameters. We present in the following lemma the gradient formulas of (5).

[1] Since our focus is disentangling, the structure learnability is an independent topic in causal discovery, which could be incorporated in future work. An ablation study is done in Appendix B regarding this assumption.

Lemma 1. *Let* $r(x,z) = q(x,z)/p(x,z)$ *and* $\mathcal{D}(x,z) = \log r(x,z)$. *Then we have*

$$\begin{aligned}
\nabla_\theta L_{\text{gen}} &= -\mathbb{E}_{z\sim p_\beta(z)}[s(x,z)\nabla_x \mathcal{D}(x,z)^\top|_{x=G_\theta(z)}\nabla_\theta G_\theta(z)], \\
\nabla_\phi L_{\text{gen}} &= \mathbb{E}_{x\sim q_x}[\nabla_z \mathcal{D}(x,z)^\top|_{z=E_\phi(x)}\nabla_\phi E_\phi(x)], \\
\nabla_\beta L_{\text{gen}} &= -\mathbb{E}_\epsilon[s(x,z)(\nabla_x \mathcal{D}(x,z)^\top \nabla_\beta G(F_\beta(\epsilon)) + \nabla_z \mathcal{D}(x,z)^\top \nabla_\beta F_\beta(\epsilon))|_{z=F_\beta(\epsilon)}^{x=G(F_\beta(\epsilon))}],
\end{aligned} \tag{7}$$

where $s(x,z) = e^{\mathcal{D}(x,z)}$ *is the scaling factor.*

We then estimate the gradients in (7) by training a discriminator D via empirical logistic regression: $\min_{D'}[\frac{1}{|S_e|}\sum_{(x,z)\in S_e}\log(1+e^{-D'(x,z)}) + \frac{1}{|S_g|}\sum_{(x,z)\in S_g}\log(1+e^{D'(x,z)})]$, where S_e and S_g are finite samples from $q(x,z)$ and $p(x,z)$ respectively, leading to a GAN approach.

Based on above, we propose Algorithm 1 to learn disentangled generative causal representation.

Algorithm 1: Disentangled gEnerative cAusal Representation (DEAR) Learning

Input: training set $\{x_1,\dots,x_N,y_1,\dots,y_{N_s}\}$, initial parameters ϕ,θ,β,ψ, batch-size n

while *not convergence* **do**
  for *multiple steps* **do**
    Sample $\{x_1,\dots,x_n\}$ from the training set, $\{\epsilon_1,\dots,\epsilon_n\}$ from $\mathcal{N}(0,I)$
    Generate from the causal prior $z_i = F_\beta(\epsilon_i), i = 1,\dots n$
    Update ψ by descending the stochastic gradient:
    $\frac{1}{n}\sum_{i=1}^n \nabla_\psi \left[\log(1+e^{-D_\psi(x_i,E_\phi(x_i))}) + \log(1+e^{D_\psi(G_\theta(z_i),z_i)})\right]$
  Sample $\{x_1,\dots,x_n,y_1,\dots,y_{n_s}\}$, $\{\epsilon_1,\dots,\epsilon_n\}$ as above; generate $z_i = F_\beta(\epsilon_i)$
  Compute θ-gradient: $-\frac{1}{n}\sum_{i=1}^n s(G_\theta(z_i),z_i)\nabla_\theta D_\psi(G_\theta(z_i),z_i)$
  Compute ϕ-gradient: $\frac{1}{n}\sum_{i=1}^n \nabla_\phi D_\psi(x_i,E_\phi(x_i)) + \frac{1}{n_s}\sum_{i=1}^{n_s}\nabla_\phi L_{\text{sup}}(\phi;x_i,y_i)$
  Compute β-gradient: $-\frac{1}{n}\sum_{i=1}^n s(G(z_i),z_i)\nabla_\beta D_\psi(G_\theta(F_\beta(\epsilon_i)),F_\beta(\epsilon_i))$
  Update parameters ϕ,θ,β using the gradients

Return: ϕ,θ,β

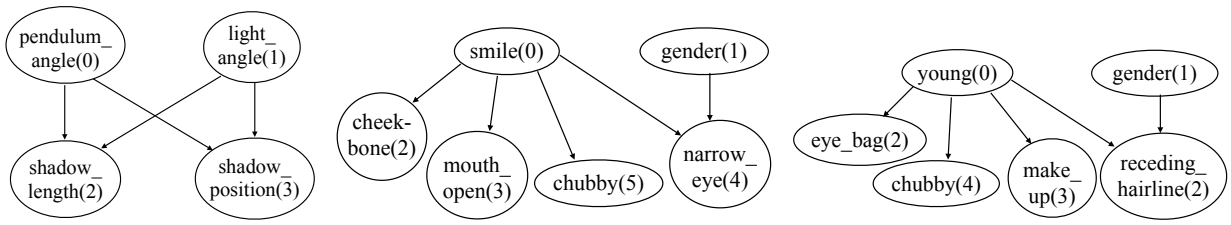
Remark: without loss of generality, assume the first N_s samples in the training set and the first n_s samples in each mini-batch has available labels; n_s may vary across different iterations.

5 Experiments

We evaluate our methods on two datasets. The first one is a synthesized dataset Pendulum similar to the one in Yang et al. (2020). As shown in Figure 3, each image is generated by four continuous factors: *pendulum_angle*, *light_angle*, *shadow_length* and *shadow_position* whose underlying structure is given in Figure 2(a) following physical mechanisms. To make the dataset realistic, we introduce random noises when generating the two effects from the causes, corresponding to the measurement error. We further introduce 20% corrupted data whose shadow is randomly generated, mimicking some environmental disturbance. The sample sizes for training, validation and test set are all 6,724.[2]

The second one is a real human face dataset CelebA (Liu et al., 2015), containing 202,599 images with 40 labelled binary attributes. Among them we consider two groups of causally related

[2]The Pendulum dataset will be released as a causal disentanglement benchmark soon. The code is available at https://github.com/xwshen51/DEAR.



(a) Pendulum (b) CelebA-Smile (c) CelebA-Attractive

Figure 2: Underlying causal structure.

factors shown in 2(b,c). We believe these two datasets are diverse enough to assess our methods. All the details of experimental setup and architectures are given in Appendix C.

5.1 Controllable generation

We first investigate the performance of our methods in disentanglement through applications in causal controllable generation (CG). Traditional CG methods mainly manipulate the independent generative factors (Karras et al., 2019), while we consider the general case where the factors are causally correlated. With a learned SCM as the prior, we are able to generate images from any desired interventional distributions of the latent factors. For example, we can manipulate only the causal factor while leave its effects unchanged. Besides, the BGM framework enables controllable generation either from scratch or a given unlabeled image.

We consider two types of intervention. In traditional traversals, we manipulate one dimension of the latent vector while keep the others fixed to either their inferred or sampled values (Higgins et al., 2017). A causal view of such operations is an intervention on all the variables by setting them as constants with only one of them varying. Another interesting type of interventional distribution is to intervene on only one latent variable, *i.e.*, $\mathbb{P}_{\mathrm{do}(Z_i=z_i)}(Z)$. The proposed SCM prior enables us to conduct such intervention though the mechanism given in Section 4.1.

Figure 3-4 illustrate the results of causal controllable generation of the proposed DEAR and the baseline method with an independent prior, S-β-VAE (Locatello et al., 2020b). Results from other baselines including S-TCVAE, S-FactorVAE (which essentially make no difference due to the independence assumption) and CausalGAN are given in Appendix D. Note that we do not compare with unsupervised disentanglement methods because of fairness and their lack of justification.

In each figure, we first infer the latent representations from a test image in block (c). The traditional traversals of two models are given in blocks (a,b). We see that in each line when manipulating one latent dimension, the generated images of our model vary only in a single factor, indicating that our method can disentangle the causally correlated factors. It is worth pointing out the we are the first to achieve the disentanglement between the cause and its effect, while other methods tend to entangle them. In block (d), we show the results of intervention on the latent variables representing the cause factors, which clearly show that intervening on a cause variable changes its effect variables. Results in Appendix D further show that intervening on an effect node does not influence its cause.

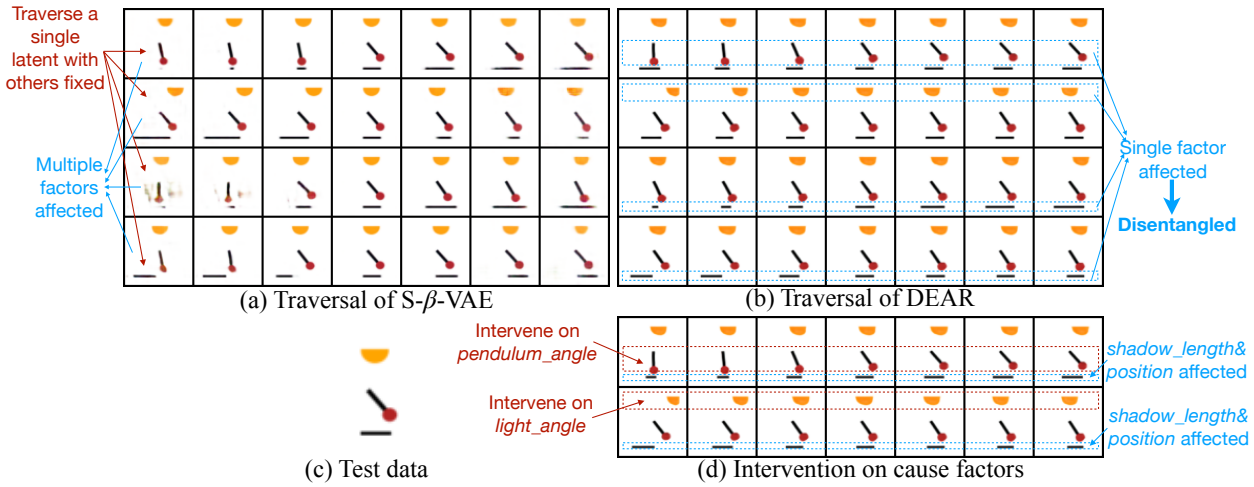


Figure 3: Results of causal controllable generation on Pendulum.

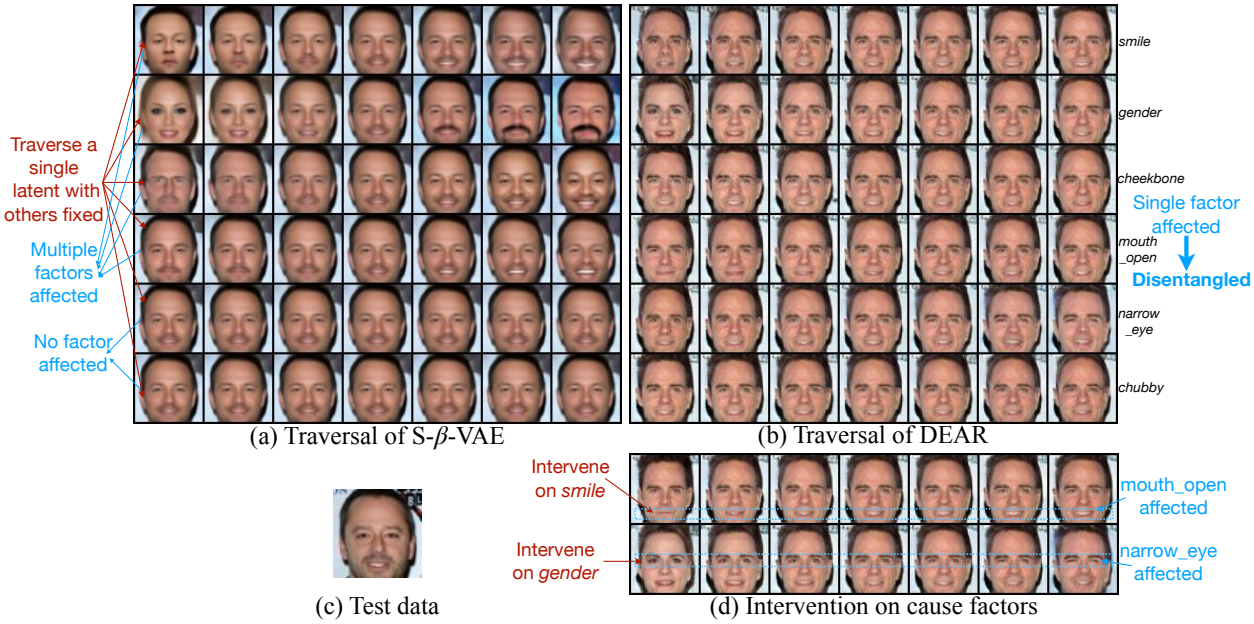


Figure 4: Results of causal controllable generation on CelebA.

Since the underlying factors are causally correlated, all previous quantitative metrics for disentanglement no longer apply. We provide more qualitative traversals in Appendix D to show the overall performance. A quantitative metric for causal disentanglement is worth exploring in future work.

5.2 Downstream task

The previous section verifies the good disentanglement performance of DEAR. In this section, equipped with DEAR, we investigate and demonstrate the benefits of learned disentangled causal

representations in sample efficiency and distributional robustness.

We state the downstream tasks. On CelebA, we consider the structure CelebA-Attractive in Figure 2(c). We artificially create a target label $\tau = 1$ if *young*=0, *gender*=0, *receding_hairline*=1, *make_up*=1, *chubby*=0, *eye_bag*=0, and $\tau = 0$ otherwise, indicating the attractiveness as a slim young woman with makeup and thick hair. On the pendulum dataset, we regard the label of data corruption as the target τ, *i.e.*, $\tau = 1$ if the data is corrupted and $\tau = 0$ otherwise. We consider the downstream tasks of predicting the target label. In both cases, the factors of interests in Figure 2(a,c) are causally related to τ, which are the features that humans use to do the task. Hence it is conjectured that a disentangled representation of these causal factors tend to be more data efficient and invariant to distribution shifts.

5.2.1 Sample efficiency

For a BGM including the previous state-of-the-art supervised disentangling methods S-VAEs (Locatello et al., 2020b) and DEAR, we use the learned encoder to embed the training data to the latent space and train a MLP classifier on the representations to predict the target label. Without an encoder, one normally needs to train a convolutional neural network with raw images as the input. Here we adopt the ResNet50 as the baseline classifier which is the architecture of the BGM encoder. Since disentangling methods use additional supervision of the generative factors, we consider another baseline that is pretrained using multi-label prediction of the factors on the same training set.

To measure the sample efficiency, we use the statistical efficiency score defined as the average test accuracy based on 100 samples divided by the average accuracy based on 10,000/all samples, following Locatello et al. (2019). Table 1 presents the results, showing that DEAR owns the highest sample efficiency on both datasets. ResNet with raw data inputs has the lowest efficiency, although multi-label pretraining improves its performance to a limited extent. S-VAEs have better efficiency than the ResNet baselines but lower accuracy under the case with more training data, which we

Table 1: Sample efficiency and test accuracy with different training sample sizes. DEAR-lin and -nlr denote the model with linear and nonlinear f. Line 1 is unsupervised; 2-3 are semi-supervised; others are supervised.

(a) CelebA

Method	**100(%)**	**10,000(%)**	**Eff(%)**
ResNet	68.06$\pm$0.19	79.51$\pm$0.31	85.59$\pm$0.27
DEAR-lin-10%	78.09$\pm$0.59	79.54$\pm$0.41	98.18$\pm$0.49
DEAR-nlr-10%	80.30$\pm$0.24	80.87$\pm$0.12	**99.29**$\pm$0.23
ResNet-pretrain	76.84$\pm$2.08	83.75$\pm$0.93	91.74$\pm$1.98
S-VAE	77.07$\pm$1.42	79.87$\pm$1.67	96.49$\pm$1.68
S-β-VAE	71.78$\pm$1.99	76.63$\pm$0.24	93.67$\pm$2.41
S-TCVAE	77.10$\pm$2.08	81.63$\pm$0.20	94.45$\pm$2.72
DEAR-lin	83.51$\pm$0.77	84.92$\pm$0.11	98.34$\pm$0.81
DEAR-nlr	**84.44**$\pm$0.48	**85.10**$\pm$0.09	**99.23**$\pm$0.51

(b) Pendulum

100(%)	**all(%)**	**Eff(%)**
79.71$\pm$0.98	90.64$\pm$1.57	87.97$\pm$2.11
88.93$\pm$1.40	93.18$\pm$0.18	95.43$\pm$1.33
87.65$\pm$0.46	91.27$\pm$0.21	96.03$\pm$0.29
79.59$\pm$0.93	89.16$\pm$1.60	89.28$\pm$0.59
84.16$\pm$0.69	90.89$\pm$0.28	92.60$\pm$0.49
79.95$\pm$1.65	87.87$\pm$0.52	90.98$\pm$1.47
85.36$\pm$1.11	90.33$\pm$0.33	94.51$\pm$1.31
90.21$\pm$0.94	**93.31**$\pm$0.14	96.68$\pm$0.89
90.62$\pm$0.32	92.57$\pm$0.08	**97.93**$\pm$0.29

think is mainly because the independent prior conflicts with the supervised loss as indicated in Proposition 1, making the learned representations entangled (as shown in the previous section) and less informative. Besides, we also investigate the performance of DEAR under the semi-supervised setting where only 10% of the labels are available. We find that DEAR with fewer labels has comparable sample efficiency with that in the fully supervised setting, with a sacrifice in accuracy that is yet still comparable to other baselines with more supervision.

5.2.2 Distributional robustness

We manipulate the training data to inject spurious correlations between the target label and some spurious attributes. On CelebA, we regard *mouth_open* as the spurious factor; on Pendulum, we choose *background_color* $\in$ {blue(+), white(−)}. We manipulate the training data such that the target label is more strongly correlated with the spurious attributes, *i.e.*, the target label and the spurious attribute of 80% of the examples are both positive or negative, while those of 20% examples are opposite. For example, in the manipulated training set, 80% smiling examples in CelebA have an open mouth; 80% corrupted examples in Pendulum are masked with a blue background. The test set however does not have these correlations, leading to a distribution shift.

Intuitively these spurious attributes are not causally correlated to the target label, but normal independent and identically distributed (IID) based methods like empirical risk minimization (ERM) tend to exploit these easily learned spurious correlations in prediction, and hence face performance degradation when the such correlation no longer exists during test. In contrast, causal factors are regarded invariant and thus robust under such shifts. Previous sections justify both theoretically and empirically that DEAR can learn disentangled causal representations. We then apply those representations by training a classifier upon them, which is conjectured to be invariant and robust. Baseline methods include ERM, multi-label ERM to predict target label and all the factors considered in disentangling to have the same amount of supervision, and S-VAEs that can not disentangle well in the causal case.

Table 2 shows the average and worst-case (Sagawa et al., 2019) test accuracy to assess both

Table 2: Distributional robustness. The worst-case and average test accuracy

	(a) CelebA		(b) Pendulum	
Method	**WorstAcc(%)**	**AvgAcc(%)**	**WorstAcc(%)**	**AvgAcc(%)**
ERM	59.12$\pm$1.78	82.12$\pm$0.26	60.48$\pm$2.73	87.40$\pm$0.89
DEAR-lin-10%	71.40$\pm$0.47	81.04$\pm$0.14	63.93$\pm$1.33	89.70$\pm$0.63
DEAR-nlr-10%	70.44$\pm$1.02	81.94$\pm$0.31	65.59$\pm$1.90	90.19$\pm$0.63
ERM-multilabel	59.17$\pm$4.02	82.05$\pm$0.25	61.70$\pm$4.02	87.20$\pm$1.00
S-VAE	60.54$\pm$3.48	79.51$\pm$0.58	20.78$\pm$4.45	84.26$\pm$1.31
S-β-VAE	63.85$\pm$2.09	80.82$\pm$0.19	44.12$\pm$9.73	86.99$\pm$1.78
S-TCVAE	64.93$\pm$3.30	81.58$\pm$0.14	35.50$\pm$5.57	86.64$\pm$1.15
DEAR-lin	**76.05**$\pm$0.70	83.56$\pm$0.09	**74.95**$\pm$1.26	**93.61**$\pm$0.13
DEAR-nlr	71.37$\pm$0.66	**83.81**$\pm$0.08	72.48$\pm$0.74	93.11$\pm$0.14

the overall classification performance and distributional robustness, where we group the test set according to the two binary labels, the target one and the spurious one, into four cases and regard the one with the worst accuracy as the worst-case, which usually owns the opposite correlation to the training data. We see that the classifiers trained upon DEAR representations outperform the baselines in both metrics. Particularly, when comparing the worst-case accuracy with the average one, we observe a slump from around 80 to around 60 for other methods on CelebA, while DEAR enjoys an acceptable small decline. These results support the above conjecture and the benefits of causal disentanglement in distributional robustness.

6 Conclusion

This paper showed that previous methods with the independent latent prior assumption fail to learn disentangled representation when the underlying factors of interests are causally correlated. We then proposed a new disentangled learning method called DEAR with theoretical guarantees. Extensive experiments demonstrated the effectiveness of DEAR in causal generation, and the benefits of the learned representations for downstream tasks.

References

Arjovsky, M., Bottou, L., Gulrajani, I., & Lopez-Paz, D. (2019). Invariant risk minimization. *arXiv preprint arXiv:1907.02893*.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1798–1828.

Bengio, Y., Deleu, T., Rahaman, N., Ke, R., Lachapelle, S., Bilaniuk, O., Goyal, A., & Pal, C. (2020). A meta-transfer objective for learning to disentangle causal mechanisms. In *ICLR*.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., & Lerchner, A. (2017). Understanding disentangling in beta-vae. *NIPS Workshop of Learning Disentangled Features*.

Chen, T. Q., Li, X., Grosse, R. B., & Duvenaud, D. K. (2018). Isolating sources of disentanglement in variational autoencoders. In *NeurIPS*.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (pp. 2172–2180).

Donahue, J., Krähenbühl, P., & Darrell, T. (2017). Adversarial feature learning. In *ICLR*.

Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., & Courville, A. C. (2017). Adversarially learned inference. In *ICLR*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672–2680).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., & Lerchner, A. (2017). beta-vae: Learning basic visual concepts with a constrained variational framework. In *ICLR*.

Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4401–4410).

Khemakhem, I., Kingma, D., Monti, R., & Hyvarinen, A. (2020). Variational autoencoders and nonlinear ica: A unifying framework. In *International Conference on Artificial Intelligence and Statistics* (pp. 2207–2217).

Kim, H. & Mnih, A. (2018). Disentangling by factorising. In *ICML*.

Kingma, D. P. & Welling, M. (2014). Auto-encoding variational bayes. In *ICLR*.

Kocaoglu, M., Snyder, C., Dimakis, A. G., & Vishwanath, S. (2018). Causalgan: Learning causal implicit generative models with adversarial training. In *ICLR*.

Kumar, A., Sattigeri, P., & Balakrishnan, A. (2018). Variational inference of disentangled latent concepts from unlabeled observations. In *ICLR*.

Leeb, F., Annadani, Y., Bauer, S., & Schölkopf, B. (2020). Structural autoencoders improve representations for generation and transfer. *arXiv preprint arXiv:2006.07796*.

Lin, Z., Thekumparampil, K. K., Fanti, G., & Oh, S. (2020). Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In *ICML*.

Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision* (pp. 3730–3738).

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., & Bachem, O. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research* (pp. 4114–4124).: PMLR.

Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., & Tschannen, M. (2020a). Weakly-supervised disentanglement without compromises. In *ICML*.

Locatello, F., Tschannen, M., Bauer, S., Rätsch, G., Schölkopf, B., & Bachem, O. (2020b). Disentangling factors of variation using few labels. In *ICLR*.

Mescheder, L., Nowozin, S., & Geiger, A. (2017). Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 2391–2400).: JMLR. org.

Mirza, M. & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Moraffah, R., Moraffah, B., Karami, M., Raglin, A., & Liu, H. (2020). Can: A causal adversarial network for learning observational and interventional distributions. *arXiv preprint arXiv:2008.11376*.

Ng, I., Ghassami, A., & Zhang, K. (2020). On the role of sparsity and dag constraints for learning linear dags. *arXiv preprint arXiv:2006.10201*.

Pearl, J. et al. (2000). Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*.

Razavi, A., van den Oord, A., & Vinyals, O. (2019). Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems* (pp. 14866–14876).

Sagawa, S., Koh, P. W., Hashimoto, T. B., & Liang, P. (2019). Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*.

Schölkopf, B. (2019). Causality for machine learning. *arXiv preprint arXiv:1911.10500*.

Schölkopf, B., Janzing, D., Peters, J., Sgouritsa, E., Zhang, K., & Mooij, J. (2012). On causal and anticausal learning. In *ICML*.

Shen, X., Zhang, T., & Chen, K. (2020). Bidirectional generative modeling using adversarial gradient estimation. *arXiv preprint arXiv:2002.09161*.

Shu, R., Chen, Y., Kumar, A., Ermon, S., & Poole, B. (2020). Weakly supervised disentanglement with guarantees. In *ICLR*.

Yang, M., Liu, F., Chen, Z., Shen, X., Hao, J., & Wang, J. (2020). Causalvae: Structured causal disentanglement in variational autoencoder. *arXiv preprint arXiv:2004.08697*.

Yu, Y., Chen, J., Gao, T., & Yu, M. (2019). Dag-gnn: Dag structure learning with graph neural networks. In *ICML*.

Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. In *International Conference on Machine Learning* (pp. 7354–7363).: PMLR.

Zhao, S., Song, J., & Ermon, S. (2017). Learning hierarchical features from generative models. In *ICML*.

Zheng, X., Aragam, B., Ravikumar, P. K., & Xing, E. P. (2018). Dags with no tears: Continuous optimization for structure learning. In *Advances in Neural Information Processing Systems* (pp. 9472–9483).

Appendix A Proofs

A.1 Proof of Proposition 1

Proof. On one hand, since E^* is disentangled wrt ξ. By Definition 1 $\forall i = 1, \ldots, m$ there exists g_i such that $E_i(x) = g_i(\xi_i)$. By the assumption that the elements of ξ are connected by a causal graph whose adjacency matrix is not a diagonal matrix. Then exist $i \neq j$ such that ξ_i and x_j are not independent, indicating that the probability density of ξ cannot be factorized.

On the other hand, notice that the distribution family of the latent prior is $\{p_z : p_z \text{ is factorized}\}$. Hence the intersection of the marginal distribution families of z and $E(x)$ is an empty set. Then the joint distribution families of $(x, E(x))$ and $(G(z), z)$ also have an empty intersection.

We know that $L_{\text{gen}}(E, G) = 0$ implies $p_E(x, z) = p_G(x, z)$ which contradicts the above. Therefore, we have $a = \min_G L_{\text{gen}}(E^*, G) > 0$.

Let (E', G') be the solution of the optimization problem $\min_{\{(E,G):L_{\text{gen}}=0\}} L_{\text{sup}}(E)$. Then we have $L' = L(E', G') = b$, and $L^* = L(E^*, G) \geq a + b^* > b^*$ for any generator G. When $b^* \geq b$ we directly have $L' < L^*$. When $b^* < b$ and λ is not large enough, i.e., $\lambda < \frac{a}{b-b^*}$, we have $L' < L^*$. □

A.2 Proof of Theorem 1

Proof. Assume E is deterministic.

On one hand, for each $i = 1, \ldots, m$, first consider the cross-entropy loss

$$L_{\text{sup},i}(E) = \mathbb{E}_{(x,y)}[\text{CE}(E_i(x), y_i)] = \int p(x)p(y_i|x)(y_i \log \sigma(E_i(x)) + (1 - y_i) \log(1 - \sigma(E_i(x))))dxdy_i.$$

Let

$$\frac{\partial L_{\text{sup},i}}{\partial \sigma(E_i(x))} = \int p(x)p(y_i|x) \left(y \frac{1}{\sigma(E_i)(1 - \sigma(E_i))} - \frac{1}{1 - \sigma(E_i)} \right) dxdy_i = 0.$$

Then we know that $E_i^*(x) = \sigma^{-1}(\mathbb{E}(y_i|x)) = \sigma^{-1}(\xi_i)$ minimizes $L_{\text{sup},i}$.

Consider the L_2 loss

$$L_{\text{sup},i}(\phi) = \mathbb{E}_{(x,y)}[\bar{E}_i(x) - y_i]^2 = \int p(x)p(y_i|x)\|E_i(x) - y_i\|^2 dxdy_i.$$

Let

$$\frac{\partial L_{\text{sup},i}}{\partial \sigma(E_i(x))} = \int p(x)p(y_i|x)(E_i(x) - y_i)dxdy_i = 0.$$

Then we know that $E_i^*(x) = \mathbb{E}(y_i|x) = \xi_i$ minimizes $L_{\text{sup},i}$ in this case.

On the other hand, we assume the infinite capacity of G and f and the learnability of the causal structure A. Thus the distribution family of $p(x, z)$ contains $q_{E^*}(x, z)$. Then we can find G^* and f^* such that $L_{\text{gen}}(E^*, G^*, f^*)$ achieves 0. Hence $L = L_{\text{gen}} + \lambda L_{\text{sup}}$ achieves minimum at $E_i^*(x) = g_i(\xi_i)$ with $g_i(\xi_i) = \sigma^{-1}(\xi_i)$ if CE loss is used, and $g_i(\xi_i) = \xi_i$ if L_2 loss is used.

For a stochastic encoder, we establish the disentanglement of its deterministic part as above, and follow Definition 1 to obtain the desired result.

□

A.3 Proof of Lemma 1

We follow the same proof scheme as in Shen et al. (2020) where the only difference lies in the gradient wrt the prior parameter β. To make this paper self-contained, we restate some proof steps here using our notations.

Let $\|\cdot\|$ denote the vector 2-norm. For a scalar function $h(x, y)$, let $\nabla_x h(x, y)$ denote its gradient with respect to x. For a vector function $g(x, y)$, let $\nabla_x g(x, y)$ denote its Jacobi matrix with respect to x. Given a differentiable vector function $g(x) : \mathbb{R}^k \to \mathbb{R}^k$, we use $\nabla \cdot g(x)$ to denote its divergence, defined as

$$\nabla \cdot g(x) := \sum_{j=1}^{k} \frac{\partial [g(x)]_j}{\partial [x]_j},$$

where $[x]_j$ denotes the j-th component of x. We know that

$$\int \nabla \cdot g(x) dx = 0$$

for all vector function $g(x)$ such that $g(\infty) = 0$. Given a matrix function $w(x) = (w_1(x), \ldots, w_l(x)) : \mathbb{R}^k \to \mathbb{R}^{k \times l}$ where each $w_i(x), i = 1 \ldots, l$ is a k-dimensional differentiable vector function, its divergence is defined as $\nabla \cdot w(x) = (\nabla \cdot w_1(x), \ldots, \nabla \cdot w_l(x))$.

To prove Lemma 1, we need the following lemma which specifies the dynamics of the generator joint distribution $p_g(x, z)$ and the encoder joint distribution $p_e(x, z)$, denoted by $p_\theta(x, z)$ and $p_\phi(x, z)$ here.

Lemma 2. *Using the definitions and notations in Lemma 1, we have*

$$\nabla_\theta p_{\theta,\beta}(x, z) = -\nabla_x p_{\theta,\beta}(x, z)^\top g_\theta(x) - p_{\theta,\beta}(x, z) \nabla \cdot g_\theta(x), \tag{8}$$

$$\nabla_\phi q_\phi(x, z) = -\nabla_z q_\phi(x, z)^\top e_\phi(z) - q_\phi(x, z) \nabla \cdot e_\phi(z), \tag{9}$$

$$\nabla_\beta p_{\theta,\beta}(x, z) = \nabla_x p_{\theta,\beta}(x, z)^\top \tilde{f}_\beta(x) - \nabla_z p_{\theta,\beta}(x, z)^\top f_\beta(z) - p_{\theta,\beta}(x, z) \nabla \cdot \begin{pmatrix} \tilde{f}_\beta(x) \\ f_\beta(z) \end{pmatrix}, \tag{10}$$

for all data x and latent variable z, where $g_\theta(G_\theta(z, \epsilon)) = \nabla_\theta G_\theta(z, \epsilon)$, $e_\phi(E_\phi(x, \epsilon)) = \nabla_\phi E_\phi(x, \epsilon)$, $f_\beta(F_\beta(\epsilon)) = \nabla_\beta F_\beta(\epsilon)$, and $\tilde{f}_\beta(G(F_\beta(\epsilon))) = \nabla_\beta G(F_\beta(\epsilon))$.

Proof of Lemma 2. We only prove (10) which is the distinct part from Shen et al. (2020).

Let l be the dimension of parameter β. To simplify notation, let random vector $Z = F_\beta(\epsilon)$ and $X = G(Z) \in \mathbb{R}^d$ and $Y = (X, Z) \in \mathbb{R}^{d+k}$, and let p be the probability density of Y. For each $i = 1, \ldots, l$, let $\Delta = \delta e_i$ where e_i is a l-dimensional unit vector whose i-th component is one and all the others are zero, and δ is a small scalar. Let $Z' = F_{\beta+\delta}(\epsilon)$, $X' = G(Z')$ and $Y' = (X', Z')$ so that Y' is a random variable transformed from Y by

$$Y' = Y + \begin{pmatrix} \tilde{f}_\beta(X) \\ f_\beta(Z) \end{pmatrix} \Delta + o(\delta).$$

Let p' be the probability density of Y'. For an arbitrary $y' = (x', z') \in \mathbb{R}^{d+k}$, let $y' = y + \binom{\tilde{f}_\beta(x)}{f_\beta(z)}\Delta + o(\delta)$ and $y = (x, z)$. Then we have

$$\begin{aligned}
p'(y') &= p(y)|\det(dy'/dy)|^{-1} \\
&= p(y)|\det(I_d + (\nabla \tilde{f}_\beta(x), \nabla f_\beta(z))^\top \Delta + o(\delta))|^{-1} \\
&= p(y)(1 + \Delta^\top \nabla \cdot (\tilde{f}_\beta(x), f_\beta(z))^\top + o(\delta))^{-1} \\
&= p(y)(1 - \Delta^\top \nabla \cdot (\tilde{f}_\beta(x), f_\beta(z))^\top + o(\delta)) \\
&= p(y) - \Delta^\top p(y') \nabla \cdot (\tilde{f}_\beta(x'), f_\beta(z'))^\top + o(\delta) \\
&= p(y') - \Delta^\top (\tilde{f}_\beta(x'), f_\beta(z')) \cdot \nabla_{x'} p(x', z) - \Delta^\top p(y') (\nabla \cdot \tilde{f}_\beta(x'), \nabla \cdot f_\beta(z'))^\top + o(\delta).
\end{aligned}$$

Since y' is arbitrary, above implies that

$$\begin{aligned}
p'(x, z) = p(x, z) &- \Delta^\top (\tilde{f}_\beta(x), f_\beta(z)) \cdot (\nabla_x p(x, z), \nabla_z p(x, z))^\top \cdot \nabla_x p(x, z) \\
&- \Delta^\top p(x, z) (\nabla \cdot \tilde{f}_\beta(x'), \nabla \cdot f_\beta(z'))^\top + o(\delta)
\end{aligned}$$

for all $x \in \mathbb{R}^d, z \in \mathbb{R}^k$ and $i = 1, \ldots, l$, leading to (10) by taking $\delta \to 0$, and noting that $p = p_\beta$ and $p' = p_{\beta+\Delta}$. Similarly we can obtain (8) and (9). □

Proof of Lemma 1. Recall the objective $D_{\mathrm{KL}}(q, p) = \int q(x, z) \log(p(x, z)/q(x, z)) dx dz$. Denote its integrand by $\ell(q, p)$. Let $\ell_2'(q, p) = \partial \ell(q, p)/\partial p$. We have

$$\nabla_\beta \ell(q(x, z), p(x, z)) = \ell_2'(q(x, z), p(x, z)) \nabla_\beta p_{\theta,\beta}(x, z)$$

where $\nabla_\beta p_{\theta,\beta}(x, z)$ is computed in Lemma 2.

Besides, we have

$$\begin{aligned}
\nabla_x \cdot [\ell_2'(q, p) p(x, z) \tilde{f}_\beta(x)] &= \ell_2'(q, p) p(x, z) \nabla \cdot \tilde{f}_\beta(x) \\
&\quad + \ell_2'(q, p) \nabla_x p(x, z) \cdot \tilde{f}_\beta(x) \\
&\quad + \nabla_x \ell_2'(q, p) p(x, z) \tilde{f}_\beta(x), \\
\nabla_z \cdot [\ell_2'(q, p) p(x, z) f_\beta(z)] &= \ell_2'(q, p) p(x, z) \nabla \cdot f_\beta(z) \\
&\quad + \ell_2'(q, p) \nabla p(x, z) \cdot f_\beta(z) \\
&\quad + \nabla \ell_2'(q, p) p(x, z) f_\beta(z).
\end{aligned}$$

Thus,

$$\nabla_\beta L_{\mathrm{gen}} = \int \nabla_\beta \ell(q(x, z), p(x, z)) dx dz = \int p(x, z) [\nabla_x \ell_2'(q, p) \tilde{f}_\beta(x) + \nabla_z \ell_2'(q, p) f_\beta(z)]$$

where we can compute $\nabla_x \ell_2'(q, p) = s(x, z) \nabla_x \mathcal{D}(x, z)$ and $\nabla_x \ell_2'(q, p) = s(x, z) \nabla_z \mathcal{D}(x, z)$.

Hence

$$\begin{aligned}
\nabla_\beta L_{\mathrm{gen}} &= -\mathbb{E}_{(x,z)\sim p(x,z)} \left[s(x, z) (\nabla_x \mathcal{D}(x, z)^\top \tilde{f}_\beta(x) + \nabla_z \mathcal{D}(x, z)^\top f_\beta(z)) \right] \\
&= -\mathbb{E}_\epsilon \left[s(x, z) (\nabla_x \mathcal{D}(x, z)^\top \nabla_\beta G(F_\beta(\epsilon)) + \nabla_z \mathcal{D}(x, z)^\top \nabla_\beta F_\beta(\epsilon)) |_{z=F_\beta(\epsilon)}^{x=G(F_\beta(\epsilon))} \right].
\end{aligned}$$

where the second equality follows reparametrization. □

Lemma 3. *For any $a, b \in \mathbb{R}$ $(a < b)$, the set of continuous piece-wise linear function P is dense in $\mathcal{C}[a,b]$ where the metric $d(f,g) = \sup_{x\in[a,b]} |f(x) - g(x)|$. Note that P is defined as*

$$P = \cup_{h\in\{(b-a)/n | n\in\mathbb{N}^+\}} P_h$$

$$P_h = \left\{ k + \sum_{i=0}^{(b-a)/h-1} w_i (x - a - ih)\mathbf{1}(x \geq a + ih) \middle| w_i, k \in \mathbb{R} \right\},$$

where $[\cdot]$ here is floor function.

Proof. Since $[a,b]$ is compact, any function $f \in \mathcal{C}[a,b]$ is uniform continuous, *i.e.,* $\forall \epsilon > 0$, there exists $\delta > 0$ such that

$$|x - y| < \delta \Longrightarrow |f(x) - f(y)| < \epsilon/2.$$

Let $[a,b] = \cup_{n=0}^{N-1} [a_n, b_n]$, and $g_n(x)$ be a linear function, such that

$$\begin{aligned} a_n &= a + nh, \\ b_n &= a + (n+1)h, \\ g_n(a_n) &= f(a_n), \\ g_i(b_n) &= f(b_n), \\ Nh &= b - a. \end{aligned}$$

Assume that $h < \delta$. For any $x \in [a_n, b_n]$, we have

$$\begin{aligned} |f(x) - g_i(x)| &\leq \min\{|f(x) - f(a_n)| + |g_i(x) - g_i(a_n)|, |f(x) - f(b_n)| + |g_i(x) - g_i(b_n)|\} \\ &\leq |g_i(a_n) - g_i(b_n)| + \min\{|f(x) - f(a_n)|, |f(x) - f(b_n)|\} \\ &\leq |f(a_n) - f(b_n)| + \min\{|f(x) - f(a_n)|, |f(x) - f(b_n)|\} \\ &< \epsilon. \end{aligned}$$

Thus,

$$\sup_{x\in[a_n,b_n]} |f(x) - g_n(x)| < \epsilon.$$

We define

$$g(x) = \sum_{n=1}^{N-1} g_n(x)\mathbf{1}(x \in [a_n, b_n])$$

which is obvious that $g(x) \in P_h \subset P$. And we have

$$\sup_{x\in[a,b]} |f(x) - g(x)| < \epsilon$$

Therefore, P is dense in $\mathcal{C}[a,b]$ and P_h is ϵ-dense.

□

Appendix B Learning the structure

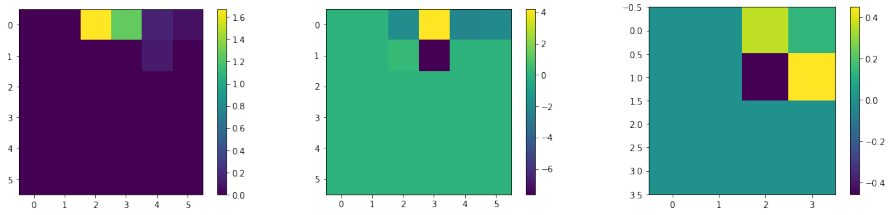
As mentioned in Section 4.1, our DEAR algorithm requires the prior knowledge on the super-graph of the true graph over the underlying factors of interests. The experiments shown in the main text are all based on the assumption that the true graph is given. In this section we investigate the performance of the learned weighted adjacency matrix and present an ablation study on different extents of prior knowledge on the structure.

B.1 Given the true graph

Figure 5 shows the learned weighted adjacency matrices when the true binary structure is given, whose weights show sensible signs and scalings consistent with common knowledge. For example, *smile* and its effect *mouth_open* are positively correlated. The corresponding element in the weighted adjacency A_{03} of (a) turns out positive, which makes sense. Also *gender* (the logit of male) and its effect *make_up* are negatively correlated. Then A_{13} of (b) turns out negative.

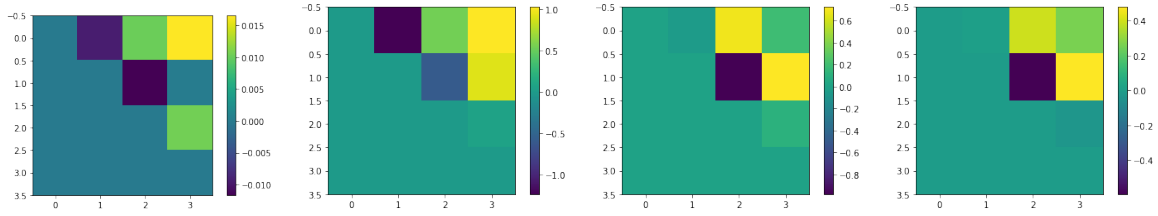
B.2 Given the true causal order

Consider the Pendulum dataset, whose ground-truth structure is given in Figure 2(a). Consider a causal order *pendulum_angle*, *light_angle*, *shadow_position*, *shadow_length*, given which we start with a full graph whose elements are randomly initialized around 0 as shown in Figure 6(a). Figure 6 presents the adjacency matrices learned by DEAR at different training epochs, from which we



(a) CelebA-Smile (b) CelebA-Attractive (c) Pendulum

Figure 5: Learned adjacency matrices for different underlying structures.



(a) Epoch 0 (b) Epoch 100 (c) Epoch 200 (d) Epoch 500

Figure 6: Learned adjacency matrices at different training epochs, starting from a random initialization.

see that it eventually obtains the learned structure that nearly coincides with the one learned given the true graph shown in Figure 5(c). This experiment shows the potential of DEAR to incorporate structure learning methods to learn the latent causal structure from scratch, which will be explored in future research.

Appendix C Implementation details

In this section we state the details of experimental setup and the network architectures used for all experiments.

Preprocessing and hyperparameters. We pre-process the images by taking a center crops of 128×128 for CelebA and resizing all images in CelebA and Pendulum to the 64×64 resolution. We adopt Adam with $\beta_1 = 0$, $\beta_2 = 0.999$, and a learning rate of 1×10^{-4} for D, 5×10^{-5} for E, G and F, and 1×10^{-3} for the adjacency matrix A. We use a mini-batch size of 128. For adversarial training in Algorithm 1, we train the D once on each mini-batch. The coefficient of the supervised regularizer is set to 5. We use CE supervised loss for both CelebA with binary observations of the underlying factors and Pendulum with bounded continuous observations. Note that L_2 loss works comparable to CE loss on Pendulum. In downstream tasks, for BGMs with an encoder, we train a two-level MLP classifier with 100 hidden nodes using Adam with a learning rate of 1×10^{-2} and a mini-batch size of 128. Models were trained for around 150 epochs on CelebA and 600 epochs on Pendulum on NVIDIA RTX 2080 Ti.

Network architectures. We follow the architectures used in Shen et al. (2020). Specifically, for such realistic data, we adopt the SAGAN (Zhang et al., 2019) architecture for D and G. The D network consists of three modules as shown in Figure 7 and detailed described in (Shen et al., 2020). Details for newtork G and D_x are given in Figure 7 and Table 3. The encoder architecture is the ResNet50 (He et al., 2016) followed by a 4-layer MLP of size 1024.

Implementation of the SCM. Recall the nonlinear SCM as the prior

$$Z = f((I - A^\top)^{-1} h(\epsilon)) := F_\beta(\epsilon).$$

We find Gaussians are expressive enough as unexplained noises, so we set h as the identity mapping. As mentioned in Section 4.1 we require the invertibility of f. We implement both linear and nonlinear ones. For a linear f, we formally refer to

$$f(z) = Wz + b,$$

where W and b are learnable weights and biases. Note that W is a diagonal matrix to model the element-wise transformation. Its inverse function can be easily computed by

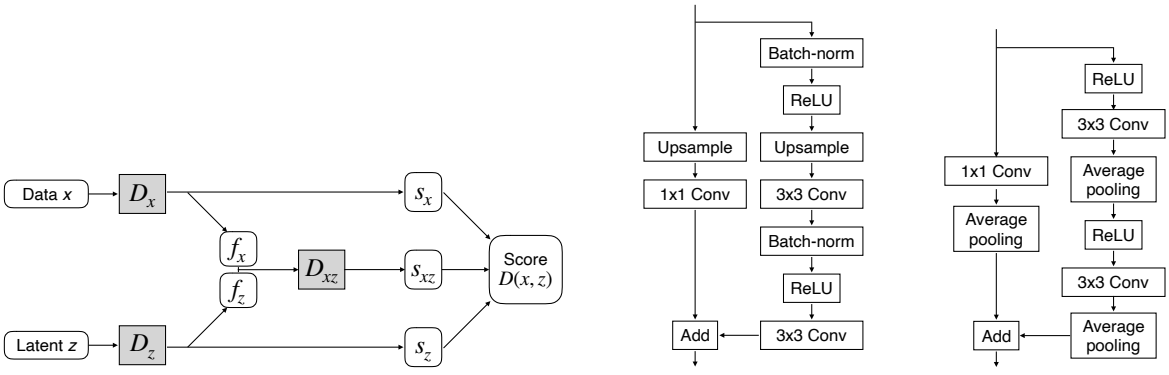
$$f^{-1}(z) = W^{-1}(z - b).$$

For a non-linear f, we use piece-wise linear functions defined by

$$f^{(i)}(z^{(i)}) = w_0^{(i)} z^{(i)} + \sum_{t=1}^{N_a} w_t^{(i)} (z^{(i)} - a_i) \mathbf{1}(z^{(i)} \geq a_i) + b^{(i)}$$

where $\cdot^{(i)}$ denote the i-th dimension of a vector or a vector-function, $a_0 < a_1 < \cdots < a_{N_a}$ are points of division, and $\mathbf{1}(\cdot)$ is the indicator function. From its denseness shown in lemma 3, the family of such piece-wise linear functions is expressive enough to model general element-wise non-linear invertible transformations.

Experimental details for baseline methods. We reproduce the S-VAEs including S-VAE, S-β-VAE and S-TCVAE using E and G with the same architecture as ours and adopt the same optimization algorithm for training. The coefficient for the independence regularizer is set to 4 since we notice that setting a larger independence regularizer hurts disentanglement in the correlated case. For the supervised regularizer, we use $\lambda = 1000$ for a balance of generative model and supervision. The ERM ResNet is trained using the same optimizer with a learning rate of 1×10^{-4}.



(a) (b) (c)

Figure 7: (a) Architecture of the discriminator $D(x, z)$; (b) A residual block (up scale) in the SAGAN generator where we use nearest neighbor interpolation for *Upsampling*; (c) A residual block (down scale) in the SAGAN discriminator.

Table 3: SAGAN architecture ($k = 100$ and $ch = 32$).

(a) Generator

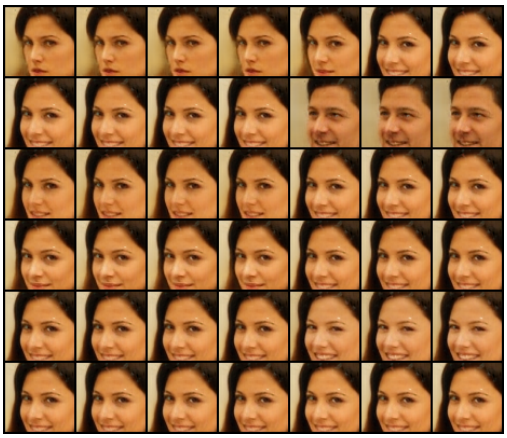
Input: $z \in \mathbb{R}^k \sim \mathcal{N}(0, I)$
Linear $\to 4 \times 4 \times 16ch$
ResBlock up $16ch \to 16ch$
ResBlock up $16ch \to 8ch$
ResBlock up $8ch \to 4ch$
Non-Local Block (64×64)
ResBlock up $4ch \to 2ch$
BN, ReLU, 3×3 Conv $2ch \to 3$
Tanh

(b) Discriminator module D_x

Input: RGB image $x \in \mathbb{R}^{64 \times 64 \times 3}$
ResBlock down $ch \to 2ch$
Non-Local Block (64×64)
ResBlock down $2ch \to 4ch$
ResBlock down $4ch \to 8ch$
ResBlock down $8ch \to 16ch$
ResBlock $16ch \to 16ch$
ReLU, Global average pooling (f_x)
Linear $\to 1$ (s_x)

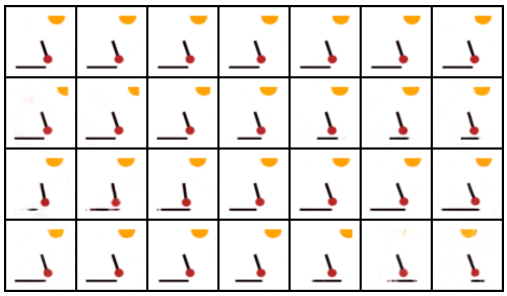
Appendix D Additional results of causal controllable generation

In this section we present more qualitative results of causal controllable generation on two datasets using DEAR and baseline methods, including S-VAEs (Locatello et al., 2020b) and CausalGAN (Kocaoglu et al., 2018). We consider three underlying structures on two datasets: Pendulum in Figure 2(a), CelebA-Smile in Figure 2(b), and CelebA-Attractive in Figure 2(c).

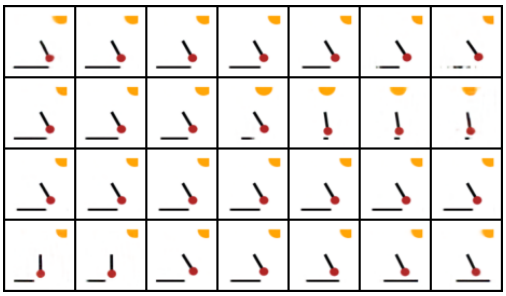


(a) CausalGAN (CelebA-Smile)

(b) CausalGAN (CelebA-Attractive)



(c) S-TCVAE (Pendulum)



(d) S-FactorVAE (Pendulum)

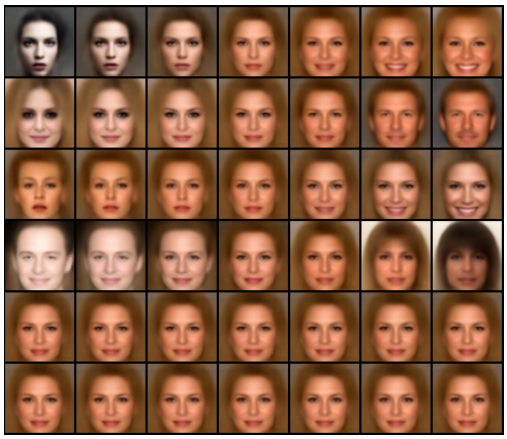
Figure 8: Traversal results of baseline methods. CausalGAN uses the binary binary factors as the conditional attributes, so the traversals appear some sudden changes. In contrast, we regard the logit of binary labels as the underlying factors and hence enjoy smooth manipulations. In addition, the controllability of CausalGAN is also limited, since entanglement still exists. Results of S-VAEs are explained in Figure 9.

(a) S-TCVAE (CelebA-Smile)

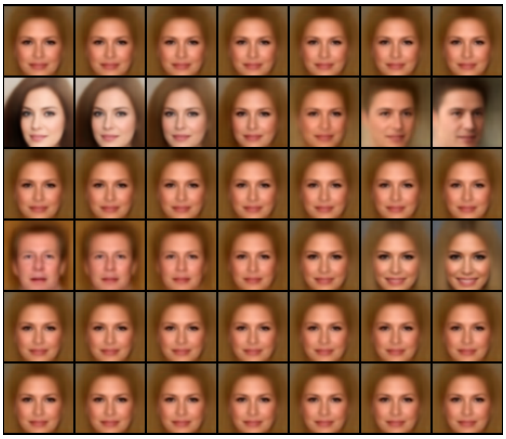
(b) S-TCVAE (CelebA-Attractive)

(c) S-FactorVAE (CelebA-Smile)

(d) S-FactorVAE (CelebA-Attractive)

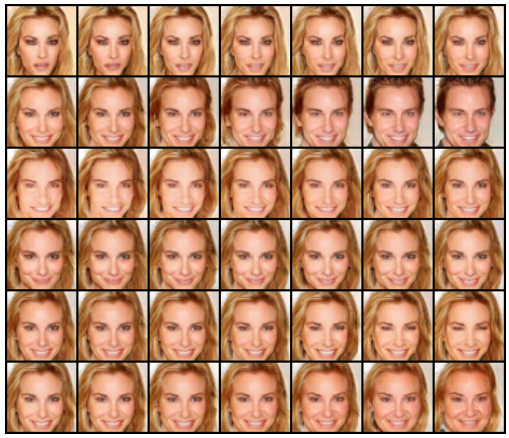


(e) S-β-VAE (CelebA-Smile)

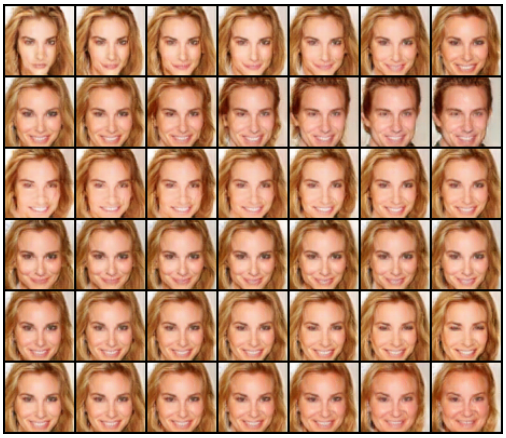


(f) S-β-VAE (CelebA-Attractive)

Figure 9: Traversal results of baseline methods. We see that (1) entanglement occurs; (2) some factors are not detected (traversing on some dimensions of the latent vector makes no difference in the decoded images.) Besides, the generated images from VAEs are blurry.



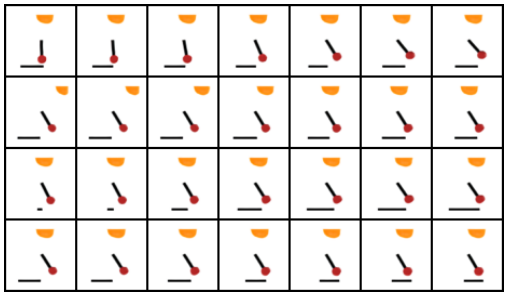
(a) Traversal (CelebA-Smile)



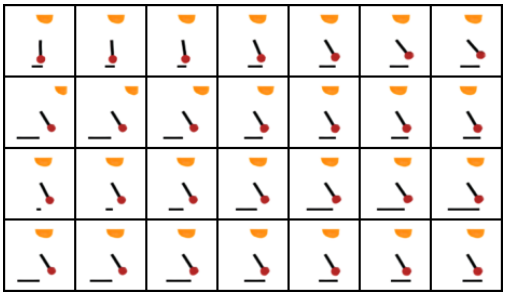
(b) Intervention (CelebA-Smile)

(c) Traversal (CelebA-Attractive)

(d) Intervention (CelebA-Attractive)



(e) Traversal (Pendulum)



(f) Intervention (Pendulum)

Figure 10: Results of DEAR. Note that the ordering of the representations matches that of the index in Figure 2. On the left we show the traditional latent traversals (the first type of intervention stated in Section 5.1). On the right we show the results of intervening on one latent variable from which we see the consequent changes of the others (the first type of intervention). Specifically intervening on the cause variable influences the effect variables while intervening on effect variables makes no difference to the causes.