Learning Interpretable Representations with Informative Entanglements

Ege Beyazit[1] , **Doruk Tuncel**[2] , **Xu Yuan**[1] , **Nian-Feng Tzeng**[1] and **Xindong Wu**[3]
[1]University of Louisiana at Lafayette, Lafayette, LA, USA
[2]Johannes Kepler University Linz, Linz, Austria
[3]Mininglamp Academy of Sciences, Beijing, China
ege93@louisiana.edu, doruktuncel@gmail.com, xu.yuan@louisiana.edu,
nianfeng.tzeng@louisiana.edu, wuxindong@mininglamp.com

Abstract

Learning interpretable representations in an unsupervised setting is an important yet a challenging task. Existing unsupervised interpretable methods focus on extracting independent salient features from data. However they miss out the fact that the entanglement of salient features may also be informative. Acknowledging these entanglements can improve the interpretability, resulting in extraction of higher quality and a wider variety of salient features. In this paper, we propose a new method to enable Generative Adversarial Networks (GANs) to discover salient features that may be entangled in an informative manner, instead of extracting only disentangled features. Specifically, we propose a regularizer to punish the disagreement between the extracted feature interactions and a given dependency structure while training. We model these interactions using a Bayesian network, estimate the maximum likelihood parameters and calculate a negative likelihood score to measure the disagreement. Upon qualitatively and quantitatively evaluating the proposed method using both synthetic and real-world datasets, we show that our proposed regularizer guides GANs to learn representations with disentanglement scores competing with the state-of-the-art, while extracting a wider variety of salient features.

1 Introduction

Deep generative models can learn to represent high-dimensional and complex distributions by leveraging large amounts of unannotated samples. However, these models typically sacrifice the interpretability of the representations learned, in favor of accuracy [Ross and Doshi-Velez, 2018]. As the application areas of generative models grow into sensitive domains such as healthcare and security, the demand for interpretable representations increases. Additionally, transfer learning, zero-shot learning and reinforcement learning methods also benefit from interpretable representations that enhance the utility of data [Kim and Mnih, 2018]. To learn interpretable representations in an unsupervised manner, a common practice is encouraging disentanglement by learning a set of independent factors of variation, corresponding to the salient features of training data [Gilpin *et al.*, 2018]. However, solely focusing on learning these independent factors oversees the fact that entanglement of some features may as well be informative. For instance, the causal relationships among the features may provide more intuitive information than only observing their independent marginals. Interactions of some features may implicitly form new higher-order features that improve the interpretability. Ideally, to achieve complete interpretability, a learner shall be able to have control over both disentangled and informatively entangled salient features. However, it is challenging to simultaneously satisfy these two constraints as most real-world datasets do not include supervision pointing out the features that are entangled but salient, or disentangled but not salient.

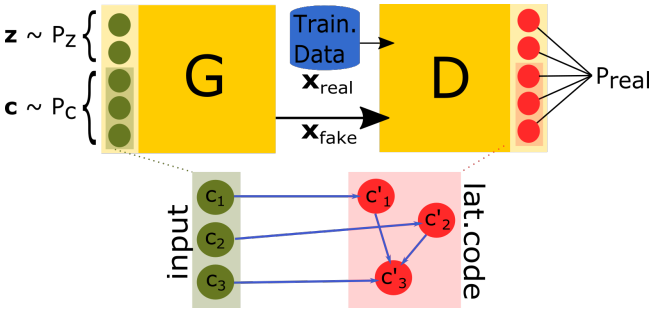


Figure 1: An example GAN with its input and latent code interactions modeled as a Bayesian network

Being two influential works with distinct approaches for generative deep learning, Variational Autoencoders and Generative Adversarial Networks have been extensively used to design generative models with the aim of learning interpretable representations. However, they focus only on disentangling, thus have no control over the characteristics of the latent features to be extracted, other than constraining them to be independent from each other. As a result, these methods inevitably fail to discover the salient features resulting from other features' interactions. In addition, they do not have control over the granularity of the extracted features, which may result in extracting independent but non-salient compositions of salient features.

In this paper, we aim to learn interpretable representations of both disentangled and entangled salient features with GANs. A regularizer is proposed to guide the learner to discover a set of features that interact according to a given dependency structure. Specifically, the relationships between the observed and latent variables are modeled by using a Bayesian network, as shown in Figure 1. Then, the difference between the structure of the Bayesian network and set of interactions among the observed and latent variables is used as a regularizer during training.

The contributions of this paper are summarized as follows:

- We propose a regularizer to impose structural constraints on the latent space variable interactions, which works effectively to explore the salient features of data.
- Our solution is shown to work on various synthetic and real-world datasets, achieving competitive disentanglement and better generalization than the state-of-the-art.
- We validate that our regularizer can discover a wider variety of salient features than the state-of-the-art methods, by considering both disentangled and informatively entangled factors.

2 Related Work

In real-world applications, supervision requires labeling, which is labor-intensive and time-consuming. Unsupervised learning methods excel at such applications by exploring hidden structures from unlabeled data [Ranzato *et al.*, 2007; Perry *et al.*, 2010]. This motivates unsupervised disentangled representation learning, in which the model aims to discover independent factors of variations [Schmidhuber, 1992; Tang *et al.*, 2013].

VAEs try to explicitly construct density functions from data by making variational approximations [Kingma and Welling, 2013]. Specifically, to make the explicit modeling tractable, VAEs define a lower bound for the log-likelihood and maximize this lower bound instead. [Higgins *et al.*, 2017] proposed β-VAE that uses an adjustable hyperparameter additional to the VAE objective. This addition helps adjusting the strength of regularization based on the KL-divergence between the observation and the variational posterior. On the other hand, [Kim and Mnih, 2018] encouraged the latent code distribution to be factorial. Specifically, the variational objective is regularized by the negative cross-entropy loss of a discriminator that tries to classify dimensionally permuted batches. [Dupont, 2018] used Gumbel-Softmax sampling and jointly modeled continuous and discrete factors for disentanglement. [Esmaeili *et al.*, 2018] proposed a two-level hierarchical objective for VAEs, by modeling the dependencies between groups of latent variables. [Adel *et al.*, 2018] proposed two interpretable learning frameworks. First, they proposed a generalized version of VAEs to be used as an interpreter for already trained models. Then, they defined a model that is optimized by simultaneously maximizing informativeness and the compression objectives.

GANs belong to the family of implicit density among the deep generative models that can learn via maximum likelihood [Goodfellow, 2016]. GANs set up a game between two players: generator and discriminator. While the generator learns producing samples with high reconstruction fidelity, discriminator learns to separate the generator output from the training data. Compared to VAEs, GANs do not need variational bounds and are proven to be asymptotically consistent [Martin and Lon, 2017]. On the other hand, GAN training requires finding the Nash equilibrium of the game between the generator and the discriminator, which is generally a more difficult task than loss minimization with VAEs. [Radford *et al.*, 2015] proposed DCGAN to bridge the gap between Convolutional Neural Networks (CNNs) and unsupervised learning. DCGAN is able to learn image representations that support basic linear algebra. [Chen *et al.*, 2016] proposed InfoGAN that regularizes GAN's adversarial loss function with the difference between the observation and the latent code. The mutual information between the subsets of observed and latent code variables has been used to regularize the objective function. Then, a variational approximation to this regularizer has been provided to facilitate implementation. It has been shown that InfoGAN's regularizer ties a subset of observed values to the salient visual characteristics of the generator output in an unsupervised way. [Kurutach *et al.*, 2018] proposed Causal InfoGAN to combine interpretable representation learning with planning. Their proposed framework learns a generative model of sequential observations, where the generative process is induced by a transition within a low-dimensional planning model.

The existing works focus on learning representations that contain independent factors of variation to achieve better disentanglement. These works share the implicit assumption that all independent factors of variation correspond to salient features of data. This assumption overlooks two major points. First, in real-life data, some salient features may be products of the interactions of others. Ignoring these interactions may result in failure to discover additional salient features, while missing the chance of gaining extra insight into data. Second, deep models can learn complex mappings to generate independent factors that are not necessarily interpretable. Therefore, the independent factors of variation learned by a deep model may not always contribute to the interpretability of the representation. These two points suggest that to achieve better interpretability, one needs to consider both disentangled and informatively entangled salient features.

3 Background: InfoGAN

Let G represent the generator component of a GAN mapping a noise vector $\mathbf{z} \in \mathbb{R}^{d_z}$ to an implicit approximation of the sample probability distribution. The noise vector $\mathbf{z}$ is typically drawn from a factored distribution such as a Gaussian with identity covariance. We refer to the training data as *real* and the instances generated by G as *fake*. Let D be the discriminator component of GAN that learns to classify the input instances as real or fake. The state-of-the-art Generative Adversarial Network for disentangled representation learning, InfoGAN [Chen *et al.*, 2016], achieves disentanglement by regularizing the GAN's adversarial loss function with the mutual information between a set of observed variables and the generator output. It uses a generator to receive a

two-part observation vector $[\mathbf{c} \in \mathbb{R}^{d_c}, \mathbf{z} \in \mathbb{R}^{d_z}]$, where $\mathbf{c}$ denotes the vector of disentangled variables, and then redefines the *minimax* game played by the D and G components as $\min_G \max_D \mathcal{L}_{Adv}(G, D) - \lambda I(\mathbf{c}; G(\mathbf{c}, \mathbf{z}))$. Here, $\mathcal{L}_{Adv}$ denotes the adversarial loss function proposed in [Goodfellow *et al.*, 2014] and $I(\mathbf{c}; G(\mathbf{c}, \mathbf{z}))$ is the mutual information between the disentangled variables and the fake instance generated by G. Mutual information maximization encourages the network to tie the disentangled variables to the generated output, forcing the generator to assign a *meaning* to these variables. Since the mutual information component is intractable to calculate, InfoGAN approximates it by maximizing a variational lower bound instead.

Even though InfoGAN is empirically shown to be able to extract and manipulate meaningful visual features unsupervised, the regularizer $I(\mathbf{c}; G(\mathbf{c}, \mathbf{z}))$ does not guarantee the independence among the discovered salient features. On the other hand in real life, these features may be interacting with each other, resulting in a side effect of one disentangled variable being able to interfere with the value of another variable. The dependency between the salient features also makes the unsupervised exploration of new features challenging, because the representation learned by the model may arbitrarily distribute the effect of an unexplored feature onto the explored ones.

4 Methodology

We study the problem of learning interpretable representations with GANs, with the joint consideration of disentangled and informatively entangled variables. We propose to model the relationship between the observation and the salient features of data using a dependency structure, and impose this structure as a constraint for GAN training.

4.1 Modeling Variable Relationships

Motivation. To impose a structured relationship between the observed variables and the salient features, we make use of the feature extraction ability of the discriminator. Note that in GAN training, the generator updates itself based on the discriminator's output. On the other hand, the discriminator learns to extract the useful features from the training data to be able to differentiate a real instance from a fake one. Therefore, the discriminator is capable of receiving an input, real or fake, and extracting its useful features in a condensed form. If we impose a structured relationship between the observed variables and the *latent code* extracted by the discriminator, the observed variables will be tied to the salient features of the training data. Figure 1 illustrates an example of the proposed model, where the green nodes represent the observed variables, the red nodes represent the latent code of the discriminator, and the graph consisting of these nodes represents the dependency structure. In the figure, the network is being guided to extract three salient features among which the two of them cause the third, represented by the edges between the red nodes. On the other hand, the edges connecting the green nodes to red ones represent the causal relationships between the observed and latent variables, letting observed variables control the salient features of generator outputs. Since we still want the GAN to generate outputs in a stochastic manner, the structured relationship only includes subsets of the green and red nodes.

The Bayesian Network Model. We represent the joint distribution of a set of observed variables and the code generated by the discriminator as a Bayesian network [Nielsen and Jensen, 2009] for the following reasons. First, a Bayesian network structure is capable of representing variable relationships in a finer grain compared to most of the independence tests [Shen *et al.*, 2019]. Also, because the representation is of a finer granularity, the amount of data needed to model the joint distribution of the variables is less than the unstructured approach. Finally, capturing the causal relationships among the salient features can improve interpretability: how some variables are *entangled* may provide additional intuition about the data, along with what the independent factors of variation represent [Lipton, 2018].

Parameter Estimation. Let $\mathbf{z}, \mathbf{c} \sim \mathcal{N}(\mathbf{0}, I)$ represent the incompressible noise vector and the disentangled variables that are expected to take on *meaning* after training, respectively. The tuple $(\mathbf{z}, \mathbf{c})$ then represents the vector of observed variables. Also let $\mathbf{c}' \in \mathbb{R}^{d_c}$ be the latent code that is generated by the discriminator D after feature extraction. Finally, we denote the sub-network of D that generates $\mathbf{c}'$ as D_{code}. A Bayesian network represents a joint probability distribution as a product of local conditional probability distributions. We start by modeling the local distributions. Let the parents of each variable c'_i in the latent code $\mathbf{c}'$ be given as $\mathbf{p}_i = \{p_{i1}, p_{i2}, ..., p_{ik}\}$. Note that the disentangled variables in $\mathbf{c}$ do not have parents since they are directly sampled from $\mathcal{N}(\mathbf{0}, I)$. On the other hand, any c_i or c'_i can be a parent since a salient feature can be in a causal relationship between an observed variable or another salient feature. Based on this observation, we formulate the local conditional probability for c'_i as follows:

$$P(c'_i | p_{i1}, p_{i2}, ..., p_{ik}) = \mathcal{N}(w_{i0} + w_{i1}p_{i1} + w_{i2}p_{i2} + ... + w_{ik}p_{ik}; \sigma_i^2) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(c'_i - \mathbf{w}_i \cdot \mathbf{p}_i)^2}{2\sigma_i^2}\right), \quad (1)$$

where $\mathbf{w}_i$ is the weight vector that represents the linear relationship between c'_i and its parents, and σ_i^2 is the variance parameter that captures the Gaussian noise around the linear relationship. Let $\mathbf{c}'_i = \{c'_i[1], c'_i[2], ..., c'_i[m]\}$ and $P_i = \{\mathbf{p}_i[1], \mathbf{p}_i[2], ..., \mathbf{p}_i[m]\}$ represent the values of disentangled variables and the parents observed in a training batch of size m, respectively. To estimate the local conditional probability parameters $(\mathbf{w}_i, \sigma_i^2)$, we define the logarithm of the likelihood function as follows:

$$\log L(\mathbf{w}_i, \sigma_i^2 : \mathbf{c}'_i, P_i) = -\frac{1}{2}\sum_m \left[\log(2\pi\sigma_i^2) + \frac{(\mathbf{c}'_i[m] - \mathbf{w}_i \cdot \mathbf{p}_i[m])^2}{\sigma_i^2}\right]. \quad (2)$$

We take the derivative of the log-likelihood with respect to w_{ij}, and set it to zero. After rearranging, we arrive at:

$$\mathbb{E}[c'_i p_{ij}] = w_{i0}\mathbb{E}[p_{ij}] + w_{i1}\mathbb{E}[p_{i1}p_{ij}] + ... + w_{ik}\mathbb{E}[p_{ik}p_{ij}]. \quad (3)$$

To get the estimate for $\mathbf{w}_i \in \mathbb{R}^{k+1}$, we repeat this procedure for each j and solve the resulting linear system of $k+1$ equations. Note that if c'_i's parents are only defined as the observed disentangled variables, the solution is straightforward since we already know that $\mathbf{c} \sim \mathcal{N}(\mathbf{0}, I)$. Else, we use the data provided in the training batch of size m to calculate the expectations in Equation (3). We get the value of each w_{ik} by solving the resulting system of linear equations. To find σ_i^2, we start by taking the derivative of the log-likelihood with respect to w_{i0} and get:

$$\mathbb{E}[c'_i] = w_{i0} + w_{i1}\mathbb{E}[p_{i1}] + ... + w_{ik}\mathbb{E}[p_{ik}]. \quad (4)$$

Taking the derivative of the log-likelihood with respect to σ_i^2, and plugging the Equations (3) and (4) in, we arrive at:

$$\sigma_i^2 = \text{Cov}[c'_i; c'_i] - \sum_{j_1}\sum_{j_2} w_{ij_1} w_{ij_2} \text{Cov}[p_{ij_1}; p_{ij_2}]. \quad (5)$$

We can now represent the joint probability distribution of the observed and latent variables as the product of local conditional factors. Specifically for a given connectivity structure $\mathcal{G}$, we represent the joint distribution parameterized by this structure and the maximum likelihood estimates of the local conditional parameters $\hat{\theta}_{\mathcal{G}} = [(\hat{\mathbf{w}}_1, \hat{\sigma}_1), ..., (\hat{\mathbf{w}}_n, \hat{\sigma}_n)]$ as follows:

$$P(\mathbf{c}, \mathbf{c}'; \mathcal{G}, \hat{\theta}_{\mathcal{G}}) = \prod_i P(c'_i | \mathbf{p}_i; \hat{\theta}_{\mathcal{G}}) P(c_i; \mathcal{N}(\mathbf{0}, 1)). \quad (6)$$

Note that by estimating the parameters of each local conditional probability, instead of directly estimating joint distribution parameters, we gain control over the importance of individual causal relationships, which will be useful while guiding the GAN training towards a desired structure.

4.2 Regularizing GANs with Structure Loss

In this section, we design a regularizer that utilizes the value taken by the likelihood function defined in Equation (2), to guide the GAN training. Since a likelihood function measures the probability of data given a model, the value this function takes when the maximum likelihood estimates plugged in provides a natural metric to measure how well $\mathcal{G}$ fits the data. However, unlike the maximum likelihood estimation procedures that try to find the best parameters to approximate an unknown data generating distribution, we manipulate the distribution itself to find the best data generator to be represented by a given $\mathcal{G}$.

Using Equations (3) and (5), we could first estimate $\hat{\theta}_{\mathcal{G}}$, then calculate the log-likelihood from Equation (6) to regularize the objective function of GAN. However, this approach introduces two problems. The first problem of this likelihood score is as follows. The stability of this approach varies based on the number of samples in a training batch because $\hat{\theta}_{\mathcal{G}}$ is estimated from a single batch. We address this problem with the following observation. The feedforward pass of a GAN can be seen as a mapping from the space of the observed variables to a decision. Similarly, a feedforward pass starting from G and ending at D_{code} defines the following mapping:

$$G \circ D_{code}(C,\, Z;\, \theta_G, \theta_D) = C', \quad (7)$$

where θ_G and θ_D are the parameters of the generator and the discriminator, respectively. This mapping suggests that the joint distribution $P(\mathbf{c}, \mathbf{c}';\, \mathcal{G}, \hat{\theta}_{\mathcal{G}})$ can be parameterized by $G \circ D_{code}$, allowing us to express the likelihood function as $L(\theta_{\mathcal{G}}, \theta_G, \theta_D : C,\, C',\, \mathcal{G})$. According to Equations (3) and (5), the maximum likelihood estimation for $\hat{\theta}_{\mathcal{G}}$ requires the knowledge of the marginal and pairwise expectations of the observed and latent variables. Since θ_G and θ_D define the mapping in Equation (7), these parameters together with $\mathbf{z}, \mathbf{c} \sim \mathcal{N}(\mathbf{0},\, I)$ contain the sufficient statistics to estimate $\hat{\theta}_{\mathcal{G}}$. Therefore, $\hat{\theta}_{\mathcal{G}}$ can be absorbed into θ_G and θ_D. Using this observation and Equation (6), we derive the following objective to directly update the data generating distribution, the GAN, towards producing data instances that fit the given graph $\mathcal{G}$:

$$\hat{\theta}_G, \hat{\theta}_D = \text{argmax}_{\theta_G, \theta_D} \log L(\theta_G, \theta_D : C,\, C',\, \mathcal{G}) = \text{argmin}_{\theta_G, \theta_D} \sum_i \sum_k \text{MSE}(C'[i], \mathbf{p}_{ik};\, \theta_G, \theta_D), \quad (8)$$

where $C'[i]$ and $\mathbf{p}_{ik}$ correspond to a single training batch of values for c'_i and the k^{th} parent of p_i respectively. Note that Equation (8) holds because maximizing the log-likelihood is equivalent to minimizing the Mean Squared Error (MSE) for linear Gaussian models [Bishop, 2006].

The second problem of the likelihood score is as follows. Even though the score calculated using Equation (6) increases if there is a causal relationship between the parents and children of $\mathcal{G}$, this score never punishes the relationships observed from data but not specified in $\mathcal{G}$. In other words, the likelihood score based regularization does not prevent the undesired causal relationships among variables. To address this, we extend Equation (8) and propose our *structure loss* as:

$$\mathcal{L}_{Str}(C,\, C',\, \mathcal{G};\, \theta_G, \theta_D) = \sum_i \sum_k \big[\text{MSE}(C'[i], \mathbf{p}_{ik};\, \theta_G, \theta_D) - \text{MSE}(C'[i], \bar{\mathbf{p}}_{ik};\, \theta_G, \theta_D)\big], \quad (9)$$

where $\bar{\mathbf{p}}_i$ represents the values taken by the variables that are not the parents of c'_i. $\mathcal{L}_{Str}$ increases if the variables are correlated with their non-parents, and it decreases if the variables are correlated with their parents. Using the proposed loss function, we regularize GAN training as:

$$\min_G \max_D \mathcal{L}_{Adv}(G, D) + \lambda\, \mathcal{L}_{Str}(C,\, C',\, \mathcal{G};\, \theta_G, \theta_D). \quad (10)$$

Regularized by our proposed structure loss, the GAN learns to represent the training data distribution while the observed variables and the generated code relationships follow the specified graphical structure. This gives us control over the interactions of extracted variables. For example, to extract latent variables that are entirely independent from each other, we can define a graph structure with one-to-one connectivity between the observed variables and the latent code. On the other hand, to extract variables that cause each other, we can also add connections between the latent variables.

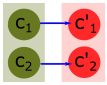
5 Experiments and Results

In this section, we conduct experiments both on synthetic and real-world datasets to evaluate the performance of our proposed regularizer. To compare our regularizer and state-of-the-art methods, the achievable disentanglement scores, quality of the latent traversals, and the variety and quality of the discovered salient features are evaluated. In our experiments, the proposed regularization is implemented on top of the same discriminator and generator architectures of InfoGAN, while tuning the parameters using grid search. The state-of-the-art methods used for comparison have been trained following the parameter and architecture settings described by their corresponding authors, unless mentioned otherwise.

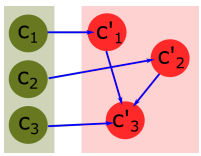
5.1 Experiments with MNIST Dataset

MNIST [LeCun *et al.*, 2010] consists of $70,000$ 28×28 grayscale images of handwritten digits, involving 10 distinct categories. Being a real-world dataset with possibly dependent natural factors, MNIST gives us the opportunity to discuss and validate our observations about both disentanglement and informative entanglement.

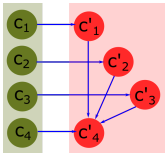
Disentanglement. To learn from MNIST, InfoGAN defines 10 categorical and 2 continuous variables to be used in $\mathbf{c}$. To train a GAN using our proposed regularizer, we set $\mathcal{G}$ to the graph structure shown in Figure 2a for the continuous random variables, and set the regularization weight λ to 0.2. To handle the categorical random variables corresponding to the digit identities, we replace the MSE function in Equation (9) with KL-Divergence. We use the same graph structure from the continuous variables, but we set the number of categorical variables in the graph to 10. Figure 3 shows the images generated by both models after training. Each row in the figure corresponds to c_i taking on values varied from -1 to 1 in evenly spaced intervals. Following observations can be made from this figure. First, even though InfoGAN captures the *rotation* feature well, the *thickness* is not sufficiently isolated as shown in Figures 3b and 3d. For almost all of the digits generated by InfoGAN, as the thickness increases, the digit also rotates. Also, the numerical identities of some digits, such as 5, are lost. On the other hand, our proposed method captures these two distinct visual features successfully without compromising the numerical identities of digits, shown in Figures 3a and 3c. We now take a closer look at the outputs from both methods to evaluate how well they generalize. Figure 4



(a) for disentanglement



(b) thickness and width isolation



(c) to discover the feature size

Figure 2: Graph structures used in experiments

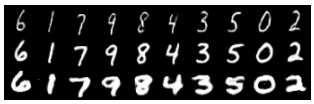
(a) c_1: Thickness (proposed)

(b) c_1: Thickness (InfoGAN)

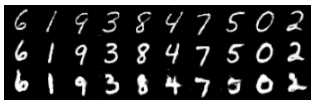
(c) c_2: Rotation (proposed)

(d) c_2: Rotation (InfoGAN)

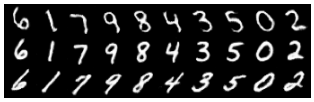
Figure 3: Latent space traversals for $c_i \in [-1, 1]$



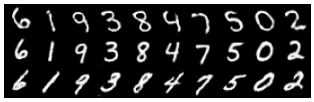
(a) Thickness (proposed)



(b) Thickness (InfoGAN)



(c) Rotation (proposed)

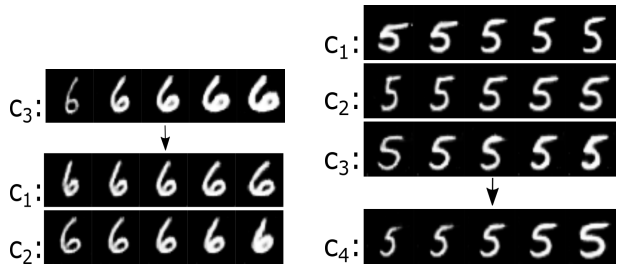


(d) Rotation (InfoGAN)

Figure 4: Generalization comparison for $c_i \in \{-2, 0, 2\}$

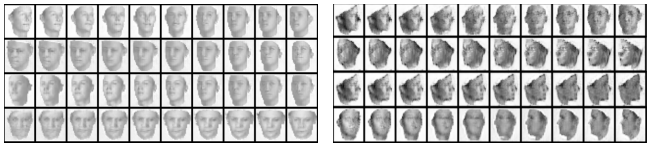
shows the images generated by setting the values of each c_i to $\{-2, 0, 2\}$. Notably, both of the models are trained using c_i values sampled from $[-1, 1]$. Hence, the quality of the output images generated by setting the c_i values outside of this range suggests how well the models are able to generalize. Comparing Figures 4a and 4b, we observe that our proposed method generalizes better than InfoGAN by generating outputs that carry better numerical identity although they become thicker. From Figure 4a, we also observe that in the representation learned using our regularizer, increasing the thickness also increases the width of the digit. This hints an existence of an informative entanglement between the width and thickness features, which we discuss in the following experiment.

Informative Entanglement. We show how our dependency structure based regularization can guide the GAN training to explore additional salient features. By exploiting the informative entanglements, it becomes possible to disentangle the features that are products of other salient features' interactions, as well as to discover new features that are entangled but salient. We start by setting $\mathcal{G}$ to the graph structure shown in Figure 2b. This graph structure regularizes the GAN towards discovering two latent features that are affecting a third one. After training the GAN, we generate samples by varying all c's from -1 to 1 in evenly spaced intervals, and show the images generated after training in Figure 5a. We observe that the variables c_1 and c_2 respectively capture the width and thickness features, while c_3 captures a mixture of width and thickness similar to our previous experiment.



(a) Width and thickness (b) Size

Figure 5: New features explored by the proposed regularizer



(a) Proposed (b) InfoGAN

Figure 6: Latent space traversals for $c_i \in [-1, 1]$ for 3D Faces

These results show that, we were able to discover two new features by modeling the entanglement between them and a third feature. To explore informative entanglements further, we repeat this experiment using the graph structure shown in Figure 2c instead. Figure 5b shows the images generated after training. We observe that the first three features captured by the model are height, width and thickness, corresponding to the variables c_1, c_2 and c_3. From the last row of the figure, we also see that the interaction of these three variables define a new entangled salient feature c_4, capturing the *size*. These experiment results suggest that the graph structure we feed to the learner, $\mathcal{G}$, can guide GANs to discover variables that follow a desired set of causal relationships.

5.2 Experiments with 3D Faces Dataset

3D faces dataset [Paysan *et al.*, 2009] contains $240,000$ face models with random variations of rotation, light, shape and elevation. Since the factors of variation of this dataset are known, we use it to demonstrate that our proposed regularizer is capable of capturing these underlying factors, while InfoGAN fails to do so. We set our method's generator and discriminator learning rates to $5e-4$ and $2e-4$ respectively. We set $\lambda = 0.1$ and $\mathcal{G}$ to the graph structure shown in Figure 2a. We set the amount of c and c' variables to 4, and extend the graph accordingly while preserving its connectivity pattern. We set the dimension of the input noise vector $\mathbf{z}$ to 52 for our proposed method, then train both models for 100 epochs. Figure 6 shows the images generated by the models, after varying each c from -1 to 1 in evenly spaced intervals. From this figure, we observe that the proposed regularizer was able to guide the GAN to represent the rotation using c_1, elevation using c_2, light using c_3 and width using c_4. On the other hand, InfoGAN only extracted three of these salient features, while the fourth feature being a mixture of the other three. This problem is caused by InfoGAN's lack of isolation among the captured salient features. Notably, InfoGAN is able to successfully capture these features using separate models with different parameter settings, but fails to capture all four using a single model. Because our regularizer punishes the similarities between the salient features, the GAN trained with it captured a wider variety of salient features from the data, without compromising on the feature isolation as much as InfoGAN does.

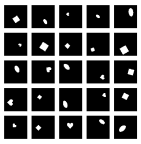


Table 1: dSprites samples

Model	Score
InfoGAN	0.820
FactorVAE	0.874
Proposed	**0.882**

Table 2: Disent. scores

5.3 Experiments with dSprites Dataset

[Matthey *et al.*, 2017] consists of $737,280$ 64×64 images of sprites, shown in Table 1, generated from known independent latent factors. This dataset has been designed to score and compare the disentanglement that different representation learning models achieve. Therefore, we use dSprites to quantitatively measure the proposed method's disentanglement capacity with the help of the disentanglement score proposed in [Kim and Mnih, 2018]. We set the weight λ of our proposed regularizer to 0.02 and, $\mathcal{G}$ to the graph structure shown in Figure 2a. We then set the amount of c and c' variables to 5, and we extend the graph accordingly while preserving its connectivity pattern. Table 2 shows that we quantitatively outperform the two state-of-the-art methods, by achieving a disentanglement score of $0.882/1.0$. This becomes possible as the proposed regularizer employs features extracted by the discriminator to represent the latent variables. While doing that, it simultaneously encourages disentanglement and discourages entanglement when $\mathcal{G}$ is set as Figure 2a.

6 Conclusion

In this paper, we have studied learning interpretable Generative Adversarial Networks by imposing structure on the explored latent feature spaces. A regularizer has been proposed by taking a graph as input and forcing a GAN to extract salient features that interact according to the graph's connectivity structure. By qualitatively and quantitatively comparing to the state-of-the-arts, we have demonstrated that our regularizer can extract additional salient features from data while achieving promising disentanglement, through imposing various constraints on the causal structure of the latent space.

Our future work includes (1) designing an algorithm that learns the optimal graph structure to explore salient features, and (2) conducting experiments with non-image datasets.

Acknowledgments

We thank IJCAI 2020 reviewers for their constructive feedback. This work is supported by the US National Science Foundation (NSF) under grants IIS-1652107 and IIS-1763620.

References

[Adel *et al.*, 2018] Tameem Adel, Zoubin Ghahramani, and Adrian Weller. Discovering interpretable representations for both deep generative and discriminative models. In *International Conference on Machine Learning*, pages 50–59, 2018.

[Bishop, 2006] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

[Dupont, 2018] Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pages 710–720, 2018.

[Esmaeili *et al.*, 2018] Babak Esmaeili, Hao Wu, Sarthak Jain, Alican Bozkurt, Narayanaswamy Siddharth, Brooks Paige, Dana H Brooks, Jennifer Dy, and Jan-Willem van de Meent. Structured disentangled representations. *arXiv preprint arXiv:1804.02086*, 2018.

[Gilpin *et al.*, 2018] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Goodfellow, 2016] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[Higgins *et al.*, 2017] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

[Kim and Mnih, 2018] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

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

[Kurutach *et al.*, 2018] Thanard Kurutach, Aviv Tamar, Ge Yang, Stuart J Russell, and Pieter Abbeel. Learning plannable representations with causal infogan. In *Advances in Neural Information Processing Systems*, pages 8733–8744, 2018.

[LeCun *et al.*, 2010] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2:18, 2010.

[Lipton, 2018] Zachary C Lipton. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

[Martin and Lon, 2017] Arjovsky Martin and B Lon. Towards principled methods for training generative adversarial networks. In *NIPS 2016 Workshop on Adversarial Training. In review for ICLR*, volume 2016, 2017.

[Matthey *et al.*, 2017] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. *URL https://github.com/deepmind/dsprites-dataset/.[Accessed on: 2018-05-08]*, 2017.

[Nielsen and Jensen, 2009] Thomas Dyhre Nielsen and Finn Verner Jensen. *Bayesian networks and decision graphs*. Springer Science & Business Media, 2009.

[Paysan *et al.*, 2009] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pages 296–301. Ieee, 2009.

[Perry *et al.*, 2010] Gavin Perry, ET Rolls, and SM Stringer. Continuous transformation learning of translation invariant representations. *Experimental brain research*, 204(2):255–270, 2010.

[Radford *et al.*, 2015] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[Ranzato *et al.*, 2007] Marc'Aurelio Ranzato, Fu Jie Huang, Y-Lan Boureau, and Yann LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *2007 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2007.

[Ross and Doshi-Velez, 2018] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[Schmidhuber, 1992] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.

[Shen *et al.*, 2019] Yujia Shen, Anchal Goyanka, Adnan Darwiche, and Arthur Choi. Structured bayesian networks: From inference to learning with routes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7957–7965, 2019.

[Tang *et al.*, 2013] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Tensor analyzers. In *International conference on machine learning*, pages 163–171, 2013.