Orthogonal Jacobian Regularization for Unsupervised Disentanglement in Image Generation

Yuxiang Wei[1,*], Yupeng Shi[1], Xiao Liu[2], Zhilong Ji[2], Yuan Gao[2], Zhongqin Wu[2], Wangmeng Zuo[1,3] (✉)
[1]Harbin Institute of Technology, [2]Tomorrow Advancing Life, [3]Pazhou Lab, Guangzhou
{yuxiang.wei.cs, csypshi}@gmail.com {liuxiao15, jizhilong, gaoyuan23, wuzhongqin}@tal.com
wmzuo@hit.edu.cn

Abstract

*Unsupervised disentanglement learning is a crucial issue for understanding and exploiting deep generative models. Recently, SeFa tries to find latent disentangled directions by performing SVD on the first projection of a pre-trained GAN. However, it is only applied to the first layer and works in a post-processing way. Hessian Penalty minimizes the off-diagonal entries of the output's Hessian matrix to facilitate disentanglement, and can be applied to multi-layers. However, it constrains each entry of output independently, making it not sufficient in disentangling the latent directions (e.g., shape, size, rotation, etc.) of spatially correlated variations. In this paper, we propose a simple **Ortho**gonal **Ja**cobian **R**egularization (**OroJaR**) to encourage deep generative model to learn disentangled representations. It simply encourages the variation of output caused by perturbations on different latent dimensions to be orthogonal, and the Jacobian with respect to the input is calculated to represent this variation. We show that our OroJaR also encourages the output's Hessian matrix to be diagonal in an indirect manner. In contrast to the Hessian Penalty, our OroJaR constrains the output in a holistic way, making it very effective in disentangling latent dimensions corresponding to spatially correlated variations. Quantitative and qualitative experimental results show that our method is effective in disentangled and controllable image generation, and performs favorably against the state-of-the-art methods. Our code is available at https://github.com/csyxwei/OroJaR.*

1. Introduction

In a disentangled representation, each dimension corresponds to the change in one factor of variation (FOV), while being independent to changes in other factors [3]. Learning disentangled representations from a given dataset is a major challenge in artificial intelligence, and can be beneficial to many computer vision tasks, such as domain adaptation [33, 45], controllable image generation [32, 38, 41, 48], and image manipulation [37].

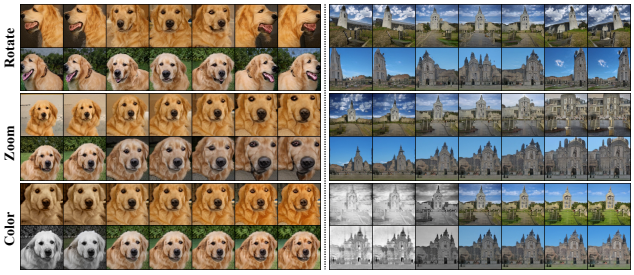


Figure 1: Examples of orthonormal directions learned by our method in BigGAN conditioned to synthesize ImageNet Golden Retrievers or Churches. Moving across a row, we move a latent code along a single linear direction in $\mathbf{z}$-space.

In the recent few years, unsupervised disentanglement learning has attracted intensive attention, owing to its importance in understanding generative models [32, 38] and extensive applications in various vision tasks [37, 45]. Based on two representative generative models, *i.e*. Variational Autoencoder (VAE) [26] and Generative Adversarial Networks (GAN) [12], many disentanglement methods [6, 7, 11, 15, 17, 25, 32, 38, 48] have been proposed. VAE-based methods, such as β-VAE [15], FactorVAE [25], β-TCVAE [6], *etc*., attain disentanglement mainly by enforcing the independence in the latent variables. However, their disentanglement performance and the visual quality of generated images remain quite limited. With the progress in Generative Adversarial Networks (GAN) [12], many GAN-based disentanglement methods have been proposed [7,32,38,48]. SeFa [38] learns the disentangled latent directions by directly decomposing the weight of the first fully-connected layer of a pre-trained GAN. However, it is only applied to the first layer of the generator model and works in a post-processing way, which limits the performance of dis-

*This work was done when Yuxiang Wei was a research intern at TAL

entanglement. Hessian Penalty [32] encourages to learn a disentangled representation by minimizing the off-diagonal entries of the output's Hessian matrix with respect to its input. However, it uses a max function to extend the regularization from scalar-valued functions to vector-valued functions, yet treats each entry of the output independently, making it not sufficient in disentangling the latent directions (*e.g.*, shape, size, rotation, *etc.*) corresponding to spatially correlated variations.

Inspired by Hessian Penalty [32] and SeFa [38], we propose a simple regularization term to encourage the generative model to learn disentangled representations. Our method is based on a straightforward intuition: when perturbing a single dimension of the network input, we would like the change in the output to be independent (and also uncorrelated) with those caused by the other input dimensions. To this end, the output's Jacobian matrix is calculated to represent the change caused by the latent input. To encourage the changes caused by different latent dimensions to be uncorrelated, we simply constrain the Jacobian vector of each dimension to be orthogonal. In contrast to Hessian Penalty, we constrain the change in a holistic way, thereby making it very competitive in disentangling latent dimensions corresponding to spatially correlated variations. We call this regularization term as **Or**th**o**gonal **Ja**cobian **R**egularization (**OroJaR**). In Sec. 3.4, we show that our OroJaR also constrains the Hessian matrix to be diagonal in an indirect way. On the other hand, our OroJaR can be treated as an end-to-end generalization of SeFa on multiple layers, which is also beneficial to the disentanglement performance. In practice, due to the fact that computing the Jacobian matrices during training is time consuming, we approximate it via a first-order finite difference approximation to accelerate training.

Experiments show that our OroJaR performs favorably against the state-of-the-art methods [32, 38, 48] for unsupervised disentanglement learning on three datasets (*i.e.*, Edges+Shoes [46], CLEVR [32], and Dsrpites [29]). Moreover, our OroJaR can be used to explore directions of meaningful variation in the latent space of pre-trained generators. From Fig. 1, our method is effective in finding the disentangled latent directions (*e.g.*, rotation, zoom and color, *etc.*) in BigGAN pre-trained on ImageNet.

The contributions of this work can be summarized as:

- We present a simple Orthogonal Jacobian Regularization (OroJaR) to encourage the deep generative model to learn better disentangled representations.
- OroJaR can be applied to multiple layers of the generator, constrains the output in a holistic way, and indirectly encourages the Hessian matrix to be diagonal.
- Extensive experiments show the effectiveness of our proposed method in learning and exploring disentangled representations, especially those corresponding to spatially correlated variations.

2. Related Work

2.1. Disentanglement Learning in VAE

Variational Autoencoder (VAE) [26] has been widely adopted in state-of-the-art disentanglement methods [6, 9, 15, 18, 21, 25, 27, 28]. β-VAE [15] introduced an adjustable hyperparameter $\beta > 1$ on the KL divergence between the variational posterior and the prior to VAE for benefiting disentangled representations, but meanwhile, it sacrificed the reconstruction result. Based on β-VAE, [25] and [6] introduced the total correlation (TC) term in order to improve disentanglement performance. DIP-VAE [27] used moment matching to penalize the divergence between aggregated posterior and the prior to encourage the disentanglement. Guided-VAE [9] used an additional discriminator to guide the unsupervised disentanglement learning and learned the latent geometric transformation and principal components. Additionally, JointVAE [11] and CascadeVAE [17] tried to simultaneously learn disentangled continuous and discrete representations in an unsupervised manner. To sum up, most existing VAE-based methods disentangle the variations mainly by factorizing aggregated posterior, but usually suffer from low-quality image generation ability.

2.2. Disentanglement Learning in GAN

Two kinds of methods, *i.e.*, two-stage and one-stage ones, have been mainly investigated for finding disentangled representations in GAN [12]. The two-stage methods identify disentangled and interpretable directions in the latent space of a pre-trained GAN. While the one-stage methods encourage disentanglement during GAN training by introducing appropriate extra regularization.

Interpretable directions in the latent space. Several unsupervised methods have been suggested for discovering interpretable directions in the latent space of a pre-trained GAN [2, 13, 37, 38, 41]. Voynov *et al.* [41] searched a set of directions that can be easily distinguished from each other by jointly learning a candidate matrix and a classifier such that the semantic directions in the matrix can be properly recognized by the classifier. Härkönen *et al.* [13] performed PCA on the sampled data to find the important and meaningful directions in the style space of StyleGAN. Shen *et al.* [38] searched the interpretable directions by performing SVD on the weight of the first layer of a pre-trained GAN. Wang *et al.* [42] unified these approaches by treating them as special cases of computing the spectrum of the Hessian for the LPIPS model [47] with respect to the input. Nonetheless, two-stage methods only work in a post-processing manner for pre-trained GANs, and generally fail to discover the disentangled components that are nonlinear in the latent space.

Disentanglement learning with regularization. Instead of post-processing, studies have also been given to achieve disentanglement by incorporating extra regularization [7, 10, 30, 32, 34, 40, 48] in GAN training. InfoGAN [7] learned the disentangled representations by maximizing the mutual information between the input latent variables and the output of the generator. Zhu *et al*. [48] presented a variation predictability loss that encourages disentanglement by maximizing the mutual information between latent variations and corresponding image pairs. Peebles *et al*. [32] proposed the Hessian Penalty to make the generator have diagonal Hessian with respect to the input. However, the max operator is used to extend Hessian Penalty for handling vector-valued output. As a result, it constrains each entry of output independently and is not sufficient in disentangling the latent directions corresponding to spatially correlated variations. Our OroJaR is motivated by Hessian Penalty [32] and SeFa [38]. It can be treated as an end-to-end generalization of SeFa to multiple layers, and constrain the change caused by latent dimension in a holistic way. Experiments also show that OroJaR is more effective in disentangling latent dimensions corresponding to spatially correlated variations.

2.3. Orthogonal Regularization

Many recent studies have been given to incorporate the orthogonality for improving deep network training [5, 19, 31, 36, 43, 44]. Wang *et al*. [43] imposed orthogonal regularization on the weighting parameters with the form $\|\mathbf{W}^T\mathbf{W}-\mathbf{I}\|_2$, where $\mathbf{W}$ is the weight matrix and $\mathbf{I}$ is an identity matrix. Jia *et al*. [19] encouraged the orthogonality by bounding the singular values of the weight matrix in a narrow range around 1. For improving image generation quality, BigGAN [4] introduced a "truncation trick" by removing the diagonal terms from the regularization. Bansal *et al*. [1] introduced another orthogonal regularization by considering both $\|\mathbf{W}^T\mathbf{W}-\mathbf{I}\|_2$ and $\|\mathbf{W}\mathbf{W}^T-\mathbf{I}\|_2$.

Besides the weight matrix, orthogonal regularization can also be used to constrain the latent space and Jacobian matrix. PrOSe [39] parameterized the latent space representation as a product of orthogonal spheres to learn disentangled representations. Odena *et al*. [31] introduced a regularization term to encourage the singular values of Jacobian matrix $\mathbf{J}$ of the generator to lie within a range. It can also constrain $\mathbf{J}$ to be orthonormal to a scale when the range is sufficiently narrow. StyleGAN2 [24] presented a path length regularization which implicitly encourages the Jacobian matrix of the generator to be orthonormal up to a global scale. While the regularizers in [24, 31] are adopted to improve the quality of the learned generator, our OroJaR is introduced to encourage the generator to learn disentangled representations. Moreover, [24, 31] encourage the Jacobian vectors to be *orthonormal* to a global scale, while our OroJaR only constrains them to be *orthogonal*.

3. Proposed Method

In this section, we first describe the proposed Orthogonal Jacobian Regularization (OroJaR) for learning disentangled representations. Then, a first-order finite difference approximation is introduced to accelerate training. Finally, we discuss its connections with the related disentanglement methods, *i.e*., SeFa [38] and Hessian Penalty [32].

3.1. Orthogonal Jacobian Regularization

Suppose G: $\mathbf{x}=G(\mathbf{z})$ is a deep generative model. Here, $\mathbf{z}=[z_1,...,z_i,...,z_m]^T\in\mathbb{R}^m$ denotes the input vector to G, and z_i denotes the i-th latent dimension. $\mathbf{x}\in\mathbb{R}^n$ denotes the output of G, and $\mathbf{x}_d=G_d(\mathbf{z})$ is further introduced to denote the the d-th layer's output of G. In terms of disentangled representation, each latent dimension is assumed to control the change in one factor of variation. That is, the changes caused by two different latent dimensions z_i and z_j should be independent (and also uncorrelated).

In our method, we use the Jacobian vector, *i.e*., $\frac{\partial G_d}{\partial z_i}$, to represent the change caused by the perturbation on the latent dimension z_i. Then, for encouraging disentangled representation, we constrain their Jacobian vectors of different latent dimensions to be orthogonal,

$$\left[\frac{\partial G_d}{\partial z_i}\right]^T\frac{\partial G_d}{\partial z_j}=0. \tag{1}$$

It is worth noting that, the orthogonality of two vectors indicates that they are uncorrelated, which also encourages the changes caused by different latent dimensions to be independent.

Taking all latent dimensions into account, we present the Orthogonal Jacobian Regularization (OroJaR) for helping deep generative model to learn disentangled representations,

$$\mathcal{L}_J(G)=\sum_{d=1}^{D}\|\mathbf{J}_d^T\mathbf{J}_d\circ(\mathbf{1}-\mathbf{I})\|=\sum_{d=1}^{D}\sum_{i=1}^{m}\sum_{j\neq i}\left|\left[\frac{\partial G_d}{\partial z_i}\right]^T\frac{\partial G_d}{\partial z_j}\right|^2, \tag{2}$$

where $\mathbf{J}_d=[\mathbf{j}_{d,1},...,\mathbf{j}_{d,i},\mathbf{j}_{d,m}]$ denotes the Jacobian matrix of G_d with respect to $\mathbf{z}$, and $\circ$ denotes the Hadamard product. $\mathbf{I}$ denotes an identity matrix, and $\mathbf{1}$ is a matrix of all ones. In particular, we use $\mathbf{j}_{d,i}=\frac{\partial G_d}{\partial z_i}$ to represent a Jacobian vector.

Our OroJaR constrains the change of output caused by latent dimension in a holistic way. To illustrate this point, we let $\mathbf{j}_d^{ij}=\mathbf{j}_{d,i}\circ\mathbf{j}_{d,j}$. Then, $\mathbf{j}_{d,i}^T\mathbf{j}_{d,j}$ can be equivalently obtained as the sum of all the elements of $\mathbf{j}_d^{ij}$. Obviously, OroJaR only constrains the summation of $\mathbf{j}_d^{ij}$ is small, and each element of $\mathbf{j}_d^{ij}$ can be positive/negative as well as large/small. Thus, our OroJaR does not impose any individual constraint on the elements of $\mathbf{j}_d^{ij}$. We note that

the changes caused by many latent semantic factors (*e.g.*, shape, size, rotation, *etc.*) usually are spatially correlated, and are better to be constrained in a holistic manner. In comparison, Hessian Penalty [32] uses a max function for aggregating the Hessian matrix of vector-valued output. It actually requires the off-diagonal entries of the Hessian matrix to be small for each element of the output, thereby making it not sufficient in disentangling the factors of complex and spatially correlated variations.

3.2. Approximation for Accelerated Training

During training, it is time consuming to compute the Jacobian matrices in Eqn. (2) when m is large. Following [16, 32], we use the Hutchinson's estimator to rewrite Eqn. (2) as:

$$\mathcal{L}_J(G) = \sum_{d=1}^{D} \mathrm{Var}_{\mathbf{v}}\left[\mathbf{v}^T(\mathbf{j}_d^T\mathbf{j}_d)\mathbf{v}\right] = \sum_{d=1}^{D} \mathrm{Var}_{\mathbf{v}}\left[(\mathbf{j}_d\mathbf{v})^T\mathbf{j}_d\mathbf{v}\right], \quad (3)$$

where $\mathbf{v}$ are Rademacher vectors (each entry has equal probability of being -1 or 1), and $\mathrm{Var}_{\mathbf{v}}$ denotes the variance. $\mathbf{j}_d\mathbf{v}$ is the first directional derivative of G in the direction $\mathbf{v}$ times $|\mathbf{v}|$. $\mathbf{j}_d\mathbf{v}$ can be efficiently computed by a first-order finite difference approximation [35]:

$$\mathbf{j}_d\mathbf{v} = \frac{1}{\epsilon}[G(\mathbf{z}+\epsilon\mathbf{v}) - G(\mathbf{z})], \quad (4)$$

where $\epsilon{>}0$ is a hyperparameter that controls the granularity of the first directional derivative estimate. In our implementation, we use $\epsilon = 0.1$.

3.3. Applications in Deep Generative Models

Our OroJaR can be applied to many generative models, and here we consider the representative Generative Adversarial Networks (GAN) [12]. The OroJaR can be applied to GAN in two ways.

Training from scratch. For GAN, the discriminator D and generator G are respectively trained using $\mathcal{L}_D$ and $\mathcal{L}_G$,

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{x}}[f(D(\mathbf{x}))] + \mathbb{E}_{\mathbf{z}}[f(1 - D(G(\mathbf{z})))], \quad (5)$$

$$\mathcal{L}_G = \mathbb{E}_{\mathbf{z}}[f(1 - D(G(\mathbf{z})))], \quad (6)$$

where f is a model-specific mapping adopted by GAN. In order to apply OroJaR to GAN training, we simply modify the loss for the generator as,

$$\mathcal{L}_G^{oro} = \mathbb{E}_{\mathbf{z}}[f(1 - D(G(\mathbf{z})))] + \lambda\mathbb{E}_{\mathbf{z}}[\mathcal{L}_J(G(\mathbf{z}))], \quad (7)$$

where λ is a trade-off hyper-parameter. Incorporating $\mathcal{L}_J(G)$ into GAN training is beneficial to learning disentangled representation, and encourages G to achieve controllable and disentangled image generation.

Apply to pre-trained generator. Analogous to Hessian Penalty [32], our OroJaR can be used to identify interpretable directions in latent space of a pretrained generator. Specifically, we introduce a learnable orthonormal matrix $\mathbf{A} \in \mathbb{R}^{m\times N}$, where N denotes the number of orthonormal directions we want to learn and m is the latent dimension; the columns of $\mathbf{A}$ store the directions we are learning. After appling the OroJaR to pre-trained G, $\mathbf{A}$ is optimized by:

$$\mathbf{A}^* = \arg\min_{\mathbf{A}} \mathbb{E}_{\mathbf{z},\omega_i}\mathcal{L}_J(G(\mathbf{z} + \eta\mathbf{A}\omega_i)), \quad (8)$$

where $\omega_i \in \{0,1\}^N$ is a one-hot vector which indexes the columns of $\mathbf{A}$ and η is a scalar which controls how far $\mathbf{z}$ should move in the direction. The difference with Eqn. (7) is the OroJaR is now taken w.r.t. ω_i instead of $\mathbf{z}$. In our training, we use $\eta = 1$. After optimization, $\mathbf{A}$ can be used to edit the generated images by $G(\mathbf{z} + \eta\mathbf{A}\omega_i)$.

3.4. Connections with SeFa and Hessian Penalty

We further discuss connections and differences of OroJaR with two representative disentanglement learning methods, *i.e.*, SeFa [38] and Hessian Penalty [32].

SeFa. SeFa [38] performs SVD on the weight matrix $\mathbf{W} \in \mathbb{R}^{m_1\times m}$ of the first layer to discover semantically meaningful directions in the latent space of pre-trained GAN. Let $\mathbf{W} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T$ be the singular value decomposition (SVD) of $\mathbf{W}$. In SeFa [38], the semantically meaningful directions are given as the column vectors of $\mathbf{V}$. We introduce $\mathbf{z}' = \mathbf{V}^T\mathbf{z}$ and $\mathbf{W}' = \mathbf{U}\mathbf{\Lambda}$, and define $G_1(\mathbf{z}) = \mathbf{W}\mathbf{z}$ and $G_1'(\mathbf{z}') = \mathbf{W}'\mathbf{z}'$. One can easily see that (i) each dimension of $\mathbf{z}'$ corresponds to a semantically meaningful direction discovered by SeFa [38]. (ii) $G_1'(\mathbf{z}')$ is equivalent with $G_1(\mathbf{z})$, *i.e.*, $G_1(\mathbf{z}) = G_1'(\mathbf{z}')$. (iii) Hard orthogonal Jacobian constraint can be attained, *i.e.*,

$$\left[\frac{\partial G_1'}{\partial z_i'}\right]^T \frac{\partial G_1'}{\partial z_j'} = 0. \quad (9)$$

Thus, SeFa [38] can be treated as a special case of our OroJaR by finding the globally optimum of $\mathcal{L}_J$ defined only on the first layer $G_1'(\mathbf{z}')$ and keeping the parameters of all other layers unchanged. In contrast to SeFa, our OroJaR can be deployed to multiple layers and be jointly optimized with GAN in an end-to-end manner, thereby being beneficial to learn better disentangled representation.

Hessian Penalty. To learn disentangled representation, Hessian Penalty [32] encourages the generator to have diagonal Hessian of the output with respect to the input. By only considering two latent dimensions z_i and z_j, the objective of Hessian Penalty can be written as,

$$\left\|\frac{\partial^2 G}{\partial z_i \partial z_j}\right\|^2 = 0. \quad (10)$$

The left term can be further decomposed into 4 components,

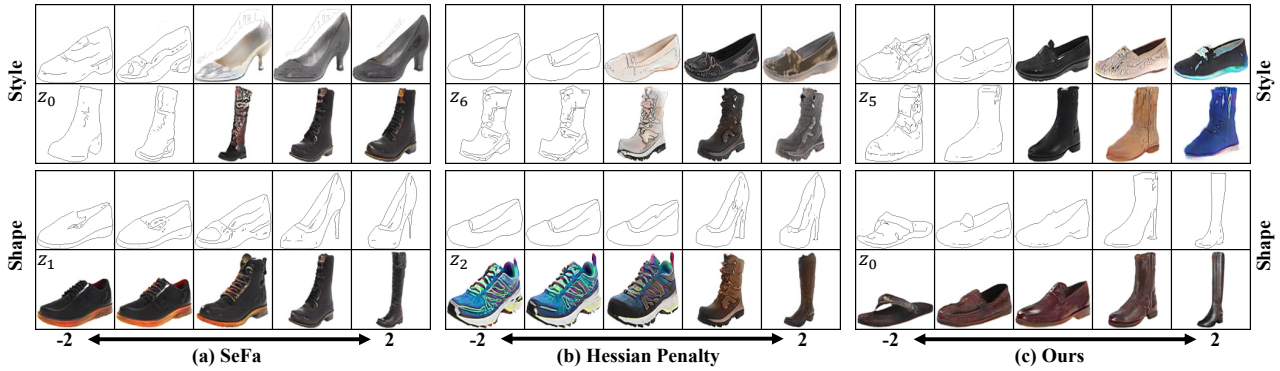


Figure 2: Comparison of disentanglement quality by our OroJaR, Hessian Penalty [32] and SeFa [38] on Edges+Shoes. For each method, we randomly sample two 12-dimensional Gaussian vectors. We select two interpretable dimensions to display, *i.e.*, the shape and style of shoes, and every two rows correspond to one interpretable dimension. Moving across a row, we vary the value of dimension z_i from -2 to $+2$ while keeping the other 11 dimensions unchanged.

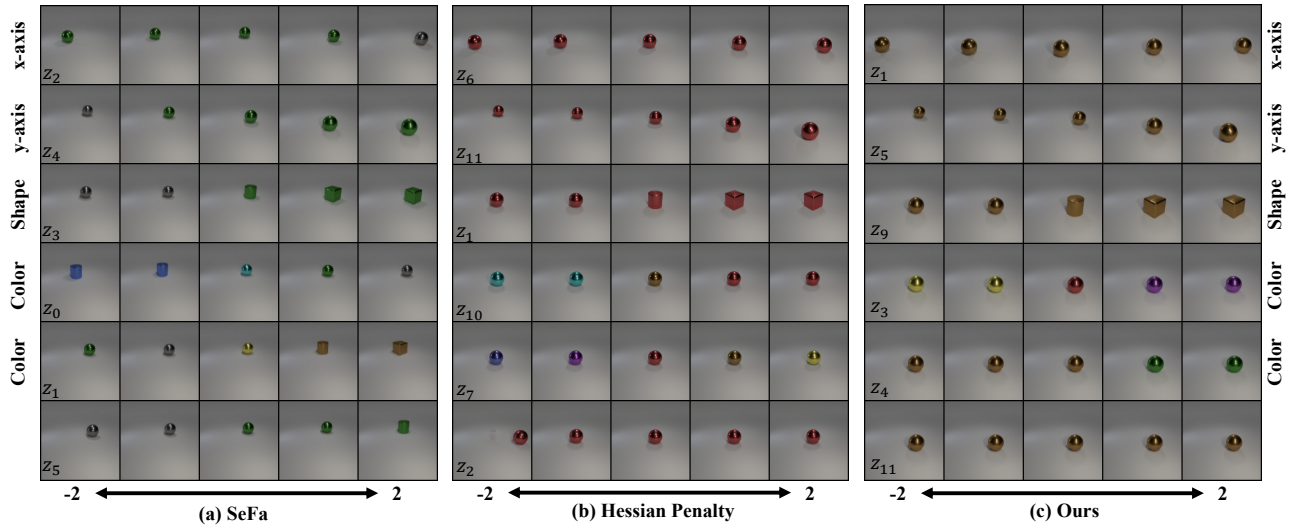


Figure 3: Comparison of disentanglement quality by our OroJaR, Hessian Penalty [32] and SeFa [38] on CLEVR-Simple. Our method has the ability to shrink the latent space when it is overparameterized. So we only show the top six activeness scoring dimensions (See Fig. 7 and Sec. 4.3). **(a)** SeFa disentangles the positions (top two rows). However it entangles the color with the shape variation (3rd-5th rows). **(b)** Hessian Penalty learns to control the vertical position, shape and color of the object independently (2nd-5th rows). However, horizontal position is unexceptedly controlled by two dimensions (1st and 6th rows). **(c)** Our method can successfully disentangle the four factors (two dimensions for color variation, but the colors controlled by them are non-overlapping) in CLEVR-Simple, and achieves better disentanglement performance.

$$
\begin{aligned}
&\left\| \frac{\partial^2 G}{\partial z_i \partial z_j} \right\|^2 = \left[\frac{\partial^2 G}{\partial z_i \partial z_j} \right]^T \frac{\partial^2 G}{\partial z_j \partial z_i} \\
\approx &\frac{1}{\delta z_i \delta z_j} \left[\frac{\partial G(z_i, z_j + \delta z_j)}{\partial z_i} \right]^T \frac{\partial G(z_i + \delta z_i, z_j)}{\partial z_j} \\
&- \frac{1}{\delta z_i \delta z_j} \left[\frac{\partial G(z_i, z_j + \delta z_j)}{\partial z_i} \right]^T \frac{\partial G(z_i, z_j)}{\partial z_j} \\
&- \frac{1}{\delta z_i \delta z_j} \left[\frac{\partial G(z_i, z_j)}{\partial z_i} \right]^T \frac{\partial G(z_i + \delta z_i, z_j)}{\partial z_j} \\
&+ \frac{1}{\delta z_i \delta z_j} \left[\frac{\partial G(z_i, z_j)}{\partial z_i} \right]^T \frac{\partial G(z_i, z_j)}{\partial z_j}.
\end{aligned} \tag{11}
$$

where $\frac{\partial G(z_i, z_j + \delta z_j)}{\partial z_i}$ is the partial gradient of G at $(z_i, z_j + \delta z_j)$ in the z_i direction, and the other items are similarly defined. When the partial gradient is smooth with the small changes in z_i and z_j, our OroJaR constrains both the last component and the other three components of Eqn. (11) to approach zero. Thus, OroJaR can offer an indirect and stronger regularization of Hessian Penalty. Moreover, OroJaR constrains the change caused by latent dimension in a holistic way, making it effective in disentangling latent dimensions corresponding to spatially correlated variations.

4. Experiments

In this section, we begin with an introduction of the datasets and implementation details, and then evaluate our OroJaR qualitatively and quantitatively by comparing it with the state-of-the-art methods. A comprehensive abla-

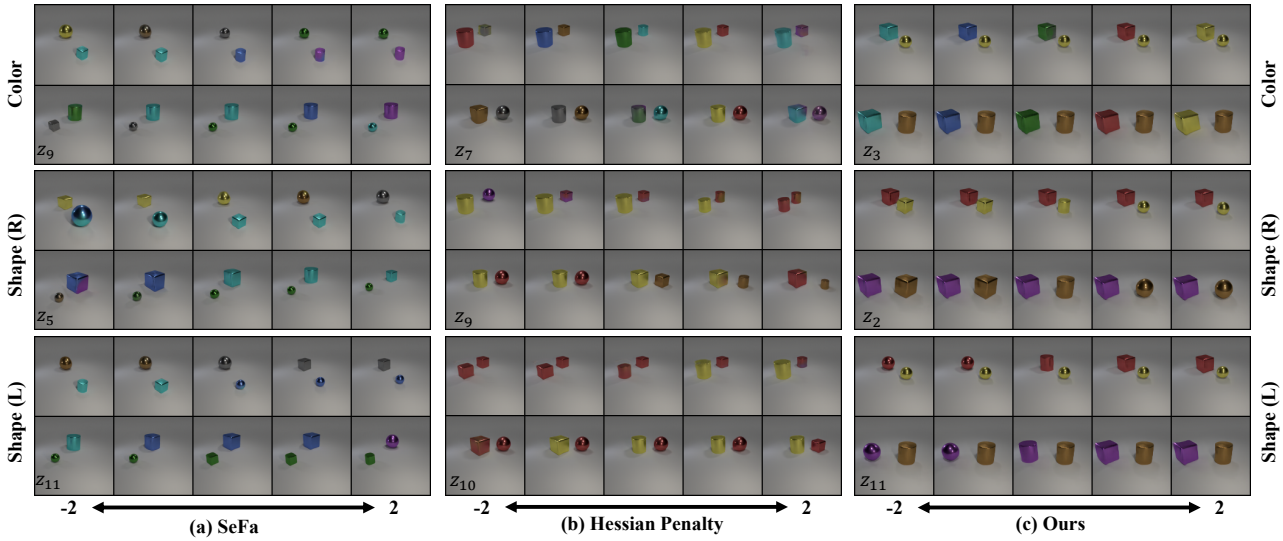


Figure 4: Comparison of disentanglement quality by our OroJaR, Hessian Penalty [32] and SeFa [38] on CLEVR-Complex. Here we show three representative factors discovered by all methods, *i.e.*, color (Top), shape of the rightmost object (Middle), and shape of the leftmost object (Bottom). **(a)** SeFa fails to disentangle the shape with color (see Middle and Bottom), and results in entangled representations. **(b)** Hessian Penalty performs poorly in controlling a single object while keeping another object unchanged. It learns to control the color of two objects by one dimension (see Top), and the shape or color of another object is also changed when changing the shape of one object (see Middle and Bottom). **(c)** Our OroJaR is effective in disentangling the color of leftmost object and the shape of each object.

tion study is given in the *suppl*.

4.1. Datasets and Implementation Details

4.1.1 Datasets

Edges+Shoes. Edges+Shoes [46] consists of 50,000 edges and 50,000 shoes images. Following [32], we adopt this dataset to evaluate whether our method can discover an independent input component to control image-to-image translation without domain supervision.

CLEVR. CLEVR dataset contians three synthetic datasets based on CLEVR [20]. The first dataset, CLEVR-1FOV, features a red cube with just a single factor of variation (FOV): object location along a single axis. The second, CLEVR-Simple, has four FOVs: object color, shape, and location (both horizontal and vertical). The third, CLEVR-Complex, retains all FOVs from CLEVR-Simple and adds a second object and another FOV (*i.e.*, object size), resulting in a total of ten FOVs (five per object). Each dataset consists of approximately 10,000 images.

Dsprites. Dsprites [29] contains totally 737,280 images generated from 5 independent latent factors (shape, size, rotation, horizontal and vertical positions).

4.1.2 Implementation Details

For Edges+Shoes and CLEVR datasets, we follow [32] to train the ProGAN [22] on them and set the dimension of input to 12. The image size is set to 128×128. For the Dsprites dataset, we train a simple GAN (6 convolution layers), and the dimension of input is set to 6. The image size is set to 64×64. In all the experiments, the OroJaR is applied right after the projection/convolution outputs for the first D (10 for ProGAN and 4 for simple GAN) layers. We find that our OroJaR empirically achieves the best disentanglement performance when D corresponds to the last layer before the last upsampling layer.

For BigGAN experiments, we set the $N = m$ and restrict $\mathbf{A}$ to be orthonormal by applying Gram-Schmidt and normalization during each forward pass.

4.2. Qualitative Evaluation

In this subsection, we qualitatively compare the disentanglement quality of our OroJaR with three state-of-the-art disentanglement methods, *i.e.*, SeFa [38], Hessian Penalty [32], and GAN-VP [48].

Edges+Shoes. Edges+Shoes dataset is a real-world but relatively simple dataset, where no ground-truth factors are provided. For a fair comparison, we choose the attributes corresponding to top two eigenvalues (lower value means ambiguous semantic direction) in SeFa. From Fig. 2, SeFa, Hessian Penalty, and our OroJaR learn the same two major disentangled variations, *i.e.*, the shape and style of shoes. While our method covers more diverse shapes.

CLEVR-Simple. Fig. 3 shows the comparison on the CLEVR-Simple dataset. We note that the number of factors in this dataset is 4, while the dimension of input is 12. When the latent space is overparameterized, our OroJaR can automatically turn off the extra dimensions. Here we only compare the top six activeness scoring dimensions with the competing methods (See Fig. 7 and Sec. 4.3). The remain-

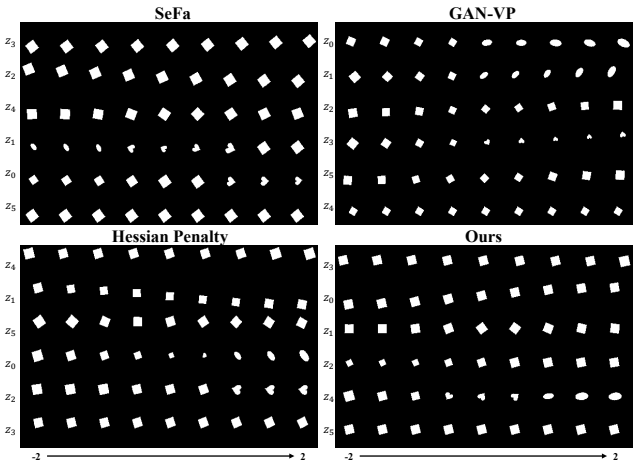


Figure 5: Comparison of disentanglement quality by SeFa [38], GAN-VP [48], Hessian Penalty [32], and our OroJaR on the Dsprites dataset. **Top-Left:** SeFa [38] entangles the rotation with the positions of object (2nd row). It also entangles the size factor with the shape factor (4th and 5th rows). **Top-Right:** For GAN-VP [48], the positions are entangled with shape and rotation. **Bottom-Left:** Hessian Penalty [32] entangles the rotation with positions, and also entangles the size with shape. **Bottom-Right:** Our method can successfully disentangle these five factors. From top to down, each row controls the horizontal position, vertical position, rotation, size, and shape, respectively. The latent dimension of the last row is correctly deactivated.

ing dimensions are deactivated based on both our OroJaR and Hessian Penalty [32], and thus are not shown. From Fig. 3, SeFa learns to control the horizontal and vertical positions of the object (top two rows), but entangles the color with the shape variations (3rd-5th rows). Hessian Penalty successfully disentangles the vertical position, shape, and color of the object (2nd-5th rows), but the horizontal position is unexpectedly controlled by two dimensions (1st and 6th rows). In comparison, our method successfully disentangles the four factors (top five rows) and deactivates the extra dimension (6th row).

CLEVR-Complex. Fig. 4 shows the comparison on the CLEVR-Complex dataset. Obviously, SeFa fails to disentangle the shape with color variations. Hessian Penalty performs poorly in controlling a single object while keeping another object unchanged. When changing the shape of one object, the shape or color of another object is also changed at the same time. A possible explanation is that Hessian Penalty constrains each entry of output independently. This makes it not sufficient in disentangling the complex latent directions (*e.g.*, shape and color of an object) corresponding to spatially correlated variations. On the contrary, our OroJaR effectively disentangles the color of the leftmost object and the shape of each object, and thus learns a better disentangled representation.

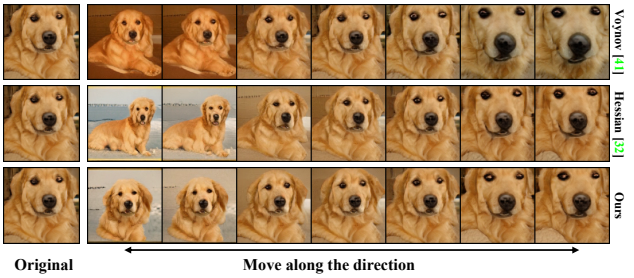


Figure 6: Comparing the quality of latent space editing by our OroJaR, Hessian Penalty [32], and Voynov [41]. The direction is added from η = -2.5 to 2.5 for Hessian Penalty and our OroJaR, and from -8 to 8 for Voynov. Our OroJaR better disentangles zoom from rotation and color.

Dsprites. Fig. 5 shows the qualitative comparison with SeFa [38], Hessian Penalty [32], and GAN-VP [48] on the Dsprites dataset. GAN-VP [48] is still limited in learning disentangled representations, where the positions are entangled with the shape and rotation. As for Hessian Penalty [32] and SeFa [38], the positions of object are entangled with the rotation. They also fail to disentangle the shape with the size variation. In contrast, our OroJaR can successfully disentangle these five factors while correctly deactivating the latent dimension of the last row. The results indicate that our OroJaR is superior in disentangling spatially correlated variations (*e.g.*, shape, size, rotation, *etc.*).

BigGAN. According to Sec. 3.3, our OroJaR can also be used to discover the meaningful latent directions of pre-trained GAN. Here we apply it to class-conditional BigGAN [4] trained on ImageNet [8]. Fig. 1 shows our results on Golden Retrievers and Churches, and our method is able to discover several disentangled directions, such as rotate, zoom, and color. Fig. 6 shows the qualitative comparison with Hessian Penalty [32] and Voynov [41]. Voynov [41] entangles the color of the dog with zoom variation. Hessian Penalty entangles the rotation with zoom variation. In contrast, our OroJaR performs a better zoom quality.

More Results. More qualitative results (*e.g.* CLEVR-U, CLEVR-1FOV, and BigGAN) are given in the suppl.

4.3. Quantitative Evaluation

In this subsection, we quantitatively compare our OroJaR with several state-of-the-art deep generative models. Following [32], we use Perceptual Path Length (PPL) and Frechet Inception Distance (FID) as the quantitative metrics. PPL [23] measures the smoothness of the generator by evaluating how much $G(\mathbf{z})$ changes under perturbations to $\mathbf{z}$. While FID [14] exploits the distance between activation distributions for measuring the quality of generated images. However, neither PPL nor FID are designed for assessing disentanglement performance. So we also report the Variation Predictability Disentanglement Metric (VP) [48] in the quantitative evaluation.

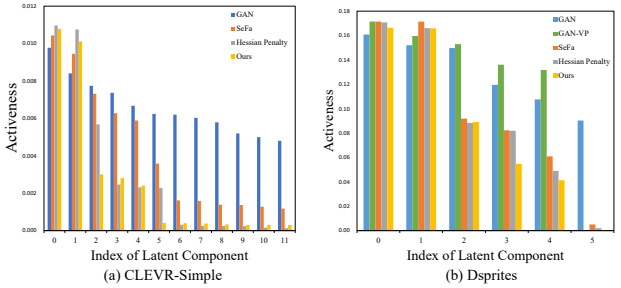
Table 1 lists the quantitative comparison results on the

Table 1: Comparison of Perceptual Path Length (PPL), Frchet Inception Distance (FID) and Variation Predictability Metric (VP) for different methods on Edges+Shoes and CLEVR. For FID and PPL, lower is better, and for VP, higher is better. We report the model with the best FID within the same number of training iterations. PPL, FID, and VP are computed with 100,000, 50,000 and 10,000 samples. The CLEVR-U dataset indicates that we train the model on CLEVR-Simple by setting $m = 3$. Due to CLEVR-1FOV only has one factor, we do not report the VP results on it.

Method	Edges+Shoes			CLEVR-Simple			CLEVR-Complex			CLEVR-U			CLEVR-1FOV		
	PPL (↓)	FID (↓)	VP (↑)	PPL	FID	VP	PPL	FID	VP	PPL	FID	VP	PPL	FID	VP
InfoGAN [7]	2952.2	**10.4**	15.6	56.2	**2.9**	28.7	83.9	**4.2**	27.9	766.7	3.6	40.1	22.1	6.2	-
ProGAN [22]	3154.1	10.8	15.5	64.5	3.8	27.2	84.4	5.5	25.5	697.7	**3.4**	40.2	30.3	9.0	-
SeFa [38]	3154.1	10.8	24.1	64.5	3.8	58.4	84.4	5.5	30.9	697.7	**3.4**	42.0	30.3	9.0	-
Hessian Penalty [32]	554.1	17.3	28.6	39.7	6.1	71.3	74.7	7.1	42.9	61.6	26.8	79.2	20.8	2.3	-
Ours	**236.7**	16.1	**32.3**	**6.7**	4.9	**76.9**	**10.4**	10.7	**48.8**	**40.9**	4.6	**90.7**	**2.8**	**2.1**	-

Table 2: Comparison of Variation Predictability Metric (VP) for different methods on Dsprites.

Method	GAN	SeFa	GAN-VP	Hessian Penalty	Ours
VP(%, ↑)	30.9 (0.84)	48.6 (0.70)	39.1 (0.48)	48.5 (0.56)	**54.7 (0.27)**



(a) CLEVR-Simple (b) Dsprites

Figure 7: Comparison of Activeness Scores (how much each dimension controls G's output) on CLEVR-Simple and Dsprites. (a) On CLEVR-Simple, both our OroJaR and Hessian Penalty [32] can deactivate the redundant dimensions (5/6 of 12 are activated). (b) On Dsprites, we have similar observation. SeFa [38] and GAN-VP [48] also have the ability to deactivate redundant dimensions.

Edges+Shoes and the CLEVR datasets. The CLEVR-1FOV dataset has only one factor and all the competing methods have the same VP value. So we do not report the VP results on this dataset. From Table 1, our OroJaR achieves better VP results on all datasets, indicating that it can learn better disentangled representation. Besides, it also serves as a path length regularization in [24] and helps learn a smooth latent space, resulting better PPL results. For our OroJaR, we empirically find that removing the normalization and activation of the first fully-connected layer is beneficial to the improvements on disentanglement. Albeit InfoGAN [7] gets lower FID on most datasets, it performs poorly in learning disentangled representation. Table 2 lists the VP results on the Dsprites dataset, and our OroJaR also achieves the highest VP among the competing methods, indicating that our OroJaR performs favorably against the state-of-the-art methods for unsupervised disentanglement learning.

In many practical scenarios, we do not have sufficient prior to setting the number of disentangled factors. One feasible solution is to use a larger dimension of input, and the disentanglement algorithm is able to identify and turn off redundant dimensions. Following [32], the activeness of a dimension z_i is introduced as the mean variance of $G(\mathbf{z})$ as we change z_i while keeping the other dimensions fixed. For assessing the ability to find redundant dimensions, Fig. 7 shows the activeness scores on CLEVR-Simple and Dsprites. In comparison to the GAN counterpart, both SeFa [38], Hessian Penalty [32], and our OroJaR is able to find redundant dimensions with smaller activeness scores. However, SeFa [38] and Hessian Penalty [32] fails to find all the redundant dimensions, which can also be observed from Fig. 3. As for GAN-VP [48], we note that the VP loss encourages the variation caused by each dimension of $\mathbf{z}$ to be distinguishable. Consequently, it can only deactivate at most one dimension, and the dimension of $\mathbf{z}$ should be carefully set to ensure GAN-VP works well. So we do not report the results of GAN-VP on Edges+Shoes and CLEVR, in which the dimension of input is set to 12 and is higher than the number of FOVs.

5. Conclusion

In this paper, we proposed an Orthogonal Jacobian Regularization (OroJaR) to help the generative model in learning disentangled representations. It encourages disentanglement by constraining the changes of output caused by different latent dimensions (*i.e.*, Jacobian vectors) to be orthogonal. Moreover, our OroJaR can be applied to multiple layers of the generator, and constrains the output in a holistic way, making it effective in disentangling latent dimensions corresponding to spatially correlated variations. Experimental results demonstrate that our OroJaR is effective in disentangled and controllable image generation, and performs favorably against the state-of-the-art methods. In the future, we will extend OroJaR to VAE and other generative models for improving disentanglement learning.

Acknowledgement

This work was supported in part by National Key R&D Program of China under Grant No. 2020AAA0104500, and by the National Natural Science Foundation of China (NSFC) under Grant No.s U19A2073 and 62006064.

References

[1] Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31:4261–4271, 2018. 3

[2] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B Tenenbaum, William T Freeman, and Antonio Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. In *International Conference on Learning Representations*, 2018. 2

[3] Yoshua Bengio. Deep learning of representations: Looking forward. In *International Conference on Statistical Language and Speech Processing*, pages 1–37. Springer, 2013. 1

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018. 3, 7

[5] Andrew Brock, Theodore Lim, James Millar Ritchie, and Nicholas J Weston. Neural photo editing with introspective adversarial networks. In *International Conference on Learning Representations*, 2017. 3

[6] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in vaes. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 2615–2625, 2018. 1, 2

[7] 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*, 2016. 1, 3, 8

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 7

[9] Zheng Ding, Yifan Xu, Weijian Xu, Gaurav Parmar, Yang Yang, Max Welling, and Zhuowen Tu. Guided variational autoencoder for disentanglement learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7920–7929, 2020. 2

[10] Chris Donahue, Zachary C Lipton, Akshay Balsubramani, and Julian McAuley. Semantically decomposing the latent spaces of generative adversarial networks. In *International Conference on Learning Representations*, 2018. 3

[11] Emilien Dupont. Learning disentangled joint continuous and discrete representations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 708–718, 2018. 1, 2

[12] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 1, 2, 4

[13] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6629–6640, 2017. 7

[15] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016. 1, 2

[16] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989. 4

[17] Yeonwoo Jeong and Hyun Oh Song. Learning discrete and continuous factors of data via alternating disentanglement. In *International Conference on Machine Learning*, pages 3091–3099. PMLR, 2019. 1, 2

[18] Ananya Harsh Jha, Saket Anand, Maneesh Singh, and VSR Veeravasarapu. Disentangling factors of variation with cycle-consistent variational auto-encoders. In *Proceedings of the European Conference on Computer Vision*, pages 805–820, 2018. 2

[19] Kui Jia, Dacheng Tao, Shenghua Gao, and Xiangmin Xu. Improving training of deep neural networks via singular value bounding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4344–4352, 2017. 3

[20] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 6

[21] Theofanis Karaletsos, Serge Belongie, and Gunnar Rätsch. Bayesian representation learning with oracle constraints. *arXiv preprint arXiv:1506.05011*, 2015. 2

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 6, 8

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 7

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 3, 8

[25] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 1, 2

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2

[27] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts

from unlabeled observations. In *International Conference on Learning Representations*, 2018. 2

[28] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019. 2

[29] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017. 2, 6

[30] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7588–7597, 2019. 3

[31] Augustus Odena, Jacob Buckman, Catherine Olsson, Tom Brown, Christopher Olah, Colin Raffel, and Ian Goodfellow. Is generator conditioning causally related to gan performance? In *International Conference on Machine Learning*, pages 3849–3858. PMLR, 2018. 3

[32] William Peebles, John Peebles, Jun-Yan Zhu, Alexei A. Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *Proceedings of the European Conference on Computer Vision*, 2020. 1, 2, 3, 4, 5, 6, 7, 8

[33] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In *International Conference on Machine Learning*, pages 5102–5112. PMLR, 2019. 1

[34] Aditya Ramesh, Youngduck Choi, and Yann LeCun. A spectral regularizer for unsupervised disentanglement. *arXiv preprint arXiv:1812.01161*, 2018. 3

[35] Clarence Hudson Richardson. *An introduction to the calculus of finite differences*. Van Nostrand, 1954. 4

[36] Pau Rodríguez, Jordi Gonzalez, Guillem Cucurull, Josep M Gonfaus, and Xavier Roca. Regularizing cnns with locally constrained decorrelations. *arXiv preprint arXiv:1611.01967*, 2016. 3

[37] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1, 2

[38] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[39] Ankita Shukla, Sarthak Bhagat, Shagun Uppal, Saket Anand, and Pavan Turaga. Product of orthogonal spheres parameterization for disentangled representation learning. *arXiv preprint arXiv:1907.09554*, 2019. 3

[40] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1415–1424, 2017. 3

[41] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International Conference on Machine Learning*, pages 9786–9796. PMLR, 2020. 1, 2, 7

[42] Binxu Wang and Carlos R Ponce. The geometry of deep generative image models and its applications. *arXiv preprint arXiv:2101.06006*, 2021. 2

[43] Daixin Wang, Peng Cui, Mingdong Ou, and Wenwu Zhu. Deep multimodal hashing with orthogonal regularization. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015. 3

[44] Jiayun Wang, Yubei Chen, Rudrasis Chakraborty, and X Yu Stella. Orthogonal convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11502–11512. IEEE Computer Society, 2020. 3

[45] Junlin Yang, Nicha C Dvornek, Fan Zhang, Julius Chapiro, MingDe Lin, and James S Duncan. Unsupervised domain adaptation via disentangled representations: Application to cross-modality liver segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 255–263. Springer, 2019. 1

[46] Aron Yu and Kristen Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 192–199, 2014. 2, 6

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 2

[48] Xinqi Zhu, Chang Xu, and Dacheng Tao. Learning disentangled representations with latent variation predictability. In *Proceedings of the European Conference on Computer Vision*, pages 684–700. Springer, 2020. 1, 2, 3, 6, 7, 8