Commutative Lie Group VAE for Disentanglement Learning

Xinqi Zhu [1] **Chang Xu** [1] **Dacheng Tao** [2]

Abstract

We view disentanglement learning as discovering an underlying structure that equivariantly reflects the factorized variations shown in data. Traditionally, such a structure is fixed to be a vector space with data variations represented by translations along individual latent dimensions. We argue this simple structure is suboptimal since it requires the model to learn to discard the properties (e.g. different scales of changes, different levels of abstractness) of data variations, which is an extra work than equivariance learning. Instead, we propose to encode the data variations with groups, a structure not only can equivariantly represent variations, but can also be adaptively optimized to preserve the properties of data variations. Considering it is hard to conduct training on group structures, we focus on Lie groups and adopt a parameterization using Lie algebra. Based on the parameterization, some disentanglement learning constraints are naturally derived. A simple model named Commutative Lie Group VAE is introduced to realize the group-based disentanglement learning. Experiments show that our model can effectively learn disentangled representations without supervision, and can achieve state-of-the-art performance without extra constraints.

1. Introduction

Equivariance has been widely considered as one of the most important desiderata in representation learning (Hinton et al., 2011; Cohen & Welling, 2014; 2016; Higgins et al., 2018). A representation is equivariant if the transformations on the input data can be reflected by transformations on the representation:

$$\sigma(g(x)) = g'\sigma(x), \tag{1}$$

[1]School of Computer Science, Faculty of Engineering, The University of Sydney, Australia [2]JD Explore Academy, JD.com, China. Correspondence to: Xinqi Zhu <xzhu7491@uni.sydney.edu.au>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

where σ denotes the representation function, and g, g' represent the same transformation acting on the data space and representation space respectively. An invariant representation is achieved if g' becomes the identity transformation.

Unsupervised disentangled representation learning is to discover the factorizable variations shown in data and encode them with individual dimensions in representations (Higgins et al., 2017; Kim & Mnih, 2018; Chen et al., 2018; Burgess et al., 2018; Jeong & Song, 2019; Zhao et al., 2017; Li et al., 2020; Karras et al., 2020). A commonly applied but less emphasized assumption is that *the disentangled representations are also equivariant*, because the semantic (or attribute) changes are reflected by the shifting of different dimensions in the representations space. The difficulty of this task is to learn the representation that preserves such an equivariance *without supervision*.

Existing unsupervised disentanglement methods usually learn the equivariance mapping based on a fixed vector space. We argue that this modeling is suboptimal, because it requires a model to do two tasks at the same time: (1) to discover equivariance; and (2) to ignore the *properties* in data variations to obey a fixed vector-space embedding. Here the *properties* in variations can include different levels of abstractness (e.g. low-level vs semantic attributes), different scales of variations (e.g. significant vs subtle changes), certain structures (e.g. cyclic), relation between variations (e.g. conditional relation), etc. Our hypothesis is that the equivariance is more likely to be learned with an adaptive equivariant structure which is used to fit the data variations. By this means, the model is relieved from doing a combined difficult learning task and can thus focus on learning equivariance. In this paper, group structures are adopted for this task, and a conceptual illustration is shown in Fig. 1. There exist some previous works that also adopt group structures to learn disentangled representations (Higgins et al., 2018; Caselles-Dupré et al., 2019; Quessard et al., 2020; Painter et al., 2020). However, these methods use predefined (known) group structures, which are neither adaptive nor generalizable. Additionally, these models cannot be learned without supervision. To the best of our knowledge, this is the first work to learn unsupervised disentangled representations based on adaptive group structures, which can be successfully applied to complex datasets.

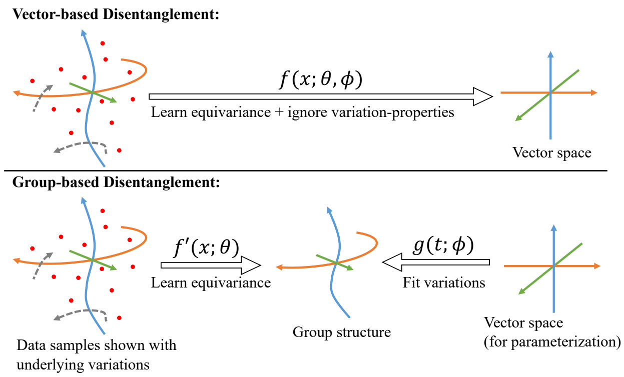

Figure 1. The upper part shows the classical disentanglement by learning equivariance with a vector space. On the left we show data samples in red points to be the static observations of some underlying variations. The underlying variations are shown as arrows in different colors with grey dash-arrows denoting noise that cannot be disentangled. The shape and length of arrows represent the properties of variations. An encoder f is trained to simultaneously learn equivariance (θ) and ignore variation-properties (ϕ). The lower part shows our proposed group-based disentanglement. Our framework separates the whole work into two parts by training an adaptive group structure to fit the variations (g) and an encoder to focus on learning equivariance (f').

In this paper, we propose to use groups *as* representations to achieve disentanglement without supervision. We focus on Lie groups for modeling continuous variations in data, and adopt the Lie algebra parameterization to enable practical training. Based on the parameterization, decomposition constraints like one-parameter subgroup decomposition and Hessian Penalty can be naturally derived to encourage disentanglement. As for the realization, we first introduce a simple variant of VAE called bottleneck-VAE, which is constructed from a new lower bound of $\log p(x)$. Based on the bottleneck-VAE, the proposed Commutative Lie Group VAE can be naturally implemented by incorporating Commutative Lie Group constraints. Our models are validated in the unsupervised disentanglement learning setting on various datasets. Without extra disentanglement constraints (like statistical independence), our group-based model achieves state-of-the-art on DSprites and 3DShapes datasets.

2. Related Work

Disentanglement Learning. Supervised disentanglement learning has been mainly tackled as conditional generation based on labeled-attributes (Reed et al., 2014; Kingma et al., 2014; Dosovitskiy et al., 2014; Kulkarni et al., 2015; Yan et al., 2016; Lample et al., 2017). After the introduction of InfoGAN (Chen et al., 2016) and β-VAE (Higgins et al., 2017), the task of unsupervised disentanglement learning has gained increasingly interest in recent years. Most of the learning models are based on the VAE framework, imposing constraints to learn continuous latent variables like statistical independence (Burgess et al., 2018; Kumar et al., 2018; Kim & Mnih, 2018; Chen et al., 2018), hierarchical biases (Chen et al., 2016; Li et al., 2020). Some other models focus on adapting discrete variables into the disentanglement learning (Dupont, 2018; Jeong & Song, 2019). Another branch of methods is based on InfoGAN, modeling the informativeness between the latent codes and the images, e.g. IB-GAN (Jeon et al., 2018), InfoGAN-CR (Lin et al., 2020), VPGAN (Zhu et al., 2020), and PS-SC GAN (Zhu et al., 2021). These models all try to capture data variations with individual latent variables in a vector space while we propose to leverage an adaptive group structure to achieve boosted disentanglement from a novel perspective.

Symmetry-Based Disentanglement Learning. In addition to the common definitions of disentanglement learning (Bengio et al., 2012; Eastwood & Williams, 2018; Do & Tran, 2020), Higgins et al. (Higgins et al., 2018) propose another formal definition of disentanglement learning based on group theory. In (Caselles-Dupré et al., 2019; Quessard et al., 2020), concrete models are proposed to learn a symmetry-based representation, but both methods rely on the paired data with action labels revealed to be trained. In (Painter et al., 2020), a reinforcement learning method is incorporated to estimate the actions, but paired data samples of elemental transformations are still required for training. These models indeed attempt to capture symmetry-variations with groups, but the group structures are predefined and the ground-truth factorization is given, leading to the inability for unsupervised disentanglement learning.

Group-Equivariant Convolutions. Inspired by the successful application of translation equivariance in convolutional neural networks (LeCun et al., 1989), there have been a large number of works trying to bring other symmetry groups into convolutional neural networks to improve data efficiency and generalization, e.g. planar rotations (Cohen & Welling, 2016; Dieleman et al., 2016; Worrall et al., 2017; Hoogeboom et al., 2018). spherical rotations (Cohen et al., 2018; Worrall & Brostow, 2018), scaling (Worrall & Welling, 2019; Sosnovik et al., 2020), general groups (Bekkers, 2020), and groups on other data structures (Finzi et al., 2020; Fuchs et al., 2020). Unlike our work, these works learn representations with certain symmetries that are predefined and usually semantic-agnostic to improve the data efficiency and generalization in neural networks. On the contrary, we focus on *discovering* variations that can be equivariantly represented and disentangled, with group structures unknown and representing semantic variations.

3. Preliminaries of Groups

This paper adopts some essential concepts from group theory which we exhibit here.

Group. A group is a set G with a binary operation $\circ$ being the group multiplication. A group satisfies the following axioms: *Closure:* For all $h, g \in G$ we have $h \circ g \in G$; *Identity:* There exists an identity element $e \in G$ such that $\forall g \in G, e \circ g = g \circ e = g$; *Inverse:* For each $g \in G$, there exists an inverse element $g^{-1} \in G$ such that $g \circ g^{-1} = g^{-1} \circ g = e$; *Associativity:* For all $g, h, i \in G$, we have $(g \circ h) \circ i = g \circ (h \circ i)$. In this paper, we consider matrix groups under matrix multiplication (a subgroup of the general linear group $GL(V)$, where V denotes the vector space on which the matrix group is acting).

Lie Group and Lie algebra. A Lie group is a group with a continuous (and smooth) structure. In this paper, we consider matrix Lie groups, which are Lie groups realized as groups of matrices. Every Lie group is associated with a Lie algebra $\mathfrak{g}$, a vector space that is the tangent space of the Lie group at the identity element. A Lie algebra can be parameterized with a basis $\{A_i\}_{i=1}^m$, where every element in $\mathfrak{g}$ is written as $A = t_1 A_1 + ... + t_m A_m$ with t_i being the coordinates. Elements in a Lie algebra can be mapped to the corresponding Lie group using the matrix exponential map $\exp : \mathfrak{g} \to G$.

One-parameter subgroup. A one-parameter subgroup is (the image of) a function $h : \mathbb{R} \to GL(V)$ if: (1) h is continuous; (2) $h(0) = e$; (3) $h(t+s) = h(t) \circ h(s)$ for all $t, s \in \mathbb{R}$. A Lie group with a 1-dim Lie algebra ($\{A_i\}_{i=1}^m = \{A\}$) is a one-parameter subgroup: $h = \exp(tA), t \in \mathbb{R}$.

Group representation. In group theory, group representations describe abstract groups as linear transformations of vector spaces. However, since we restrict our attention on matrix Lie groups and also due to its potential confusion with the term *representation* in machine learning, we discard its group-theory definition in this paper. Instead, we use the phrase *group representation* to describe a representation that is learned with a group structure.

4. Method

We propose group representations for disentanglement learning in Sec. 4.1. To enable practical learning on group representations, we introduce the Lie Algebra parameterization in Sec. 4.2. In Sec. 4.3, we first introduce a variant of VAE called bottleneck-VAE, and then introduce our Commutative Lie Group VAE by integrating the Commutative Lie Group constraints.

4.1. Group Representation

Generally in an equivariance representation (as in Eq. 1), there exist two separate components to be learned: the group (of transformations) and the representation (of data). This is not desirable since usually we need paired data to train the group as we need supervision for transformations (Caselles-Dupré et al., 2019; Quessard et al., 2020). However, if all the variations shown in a dataset can be assumed to be represented by a group (which is the case in disentanglement learning), then we can assume there exists a canonical data point x_0 such that every other data point x is transformed by a group element $g_{0\to x}$ on the canonical data point:

$$\sigma(x) = \sigma(g_{0\to x}(x_0)) = g'_{0\to x}\sigma(x_0). \tag{2}$$

Now we can see the representation structure is actually determined by the group since the group defines the *relation* between samples in the latent space while the absolute embedding position of a single data point is not important. It is thus reasonable to assign a fixed representation to the canonical data point so that only the group structure is learned. The key step is that we choose the group identity e as the canonical representation so that every sample has a *group representation*:

$$\sigma(x) = g'_{0\to x}\sigma(x_0) = g'_{0\to x}e = g'_{0\to x}, \tag{3}$$

and the samples are embedded on a group structure. This group can now be learned with static observations since each observation represents a transformation from the canonical data point. Note that this group representation is similar to the pose representation used in (Hinton et al., 2011) where the pose of a capsule is represented by a matrix which specifies the transformation between the canonical entity and the actual instantiation.

Unfortunately, though we can assume such a group representation exists, it is not clear how it can be learned in practice. It is also unclear how data samples can be mapped onto a group, how sampling can be conducted on a group, and how optimization can be implemented on a group. In this paper, we shrink our concentration on continuous groups (Lie groups) since (1) most attributes in data consist of continuous variations, and (2) it is easier to be parameterized (and thus learned). We refer this type of representations as Lie group representations.

4.2. Decomposition with Lie Algebra Parameterization

We parameterize Lie groups in this paper by a basis $\{A_i\}_{i=1}^m$ in the Lie algebra:

$$\begin{aligned} &g(t) = \exp(A(t)), g \in G, A \in \mathfrak{g}, \\ &A(t) = t_1 A_1 + t_2 A_2 + ... + t_m A_m, \forall t_i \in \mathbb{R}, \end{aligned} \tag{4}$$

where $\mathfrak{g}$ is the Lie algebra of Lie group G, and $\exp(\cdot)$ is the matrix exponential map which maps an element in a Lie algebra to the corresponding Lie group. $t = (t_1, t_2, ..., t_m)$ represents the coordinates in $\mathfrak{g}$ of a data sample. When $t = \mathbf{0}$, the corresponding element on the group is $e = \exp(\mathbf{0})$, the identity. The Lie algebra is a vector space thus enables training with general optimization methods like SGD. In our implementation, the basis $\{A_i\}_{i=1}^m$ is optimized (as weights) to find an adaptive group structure, and every data sample can thus be identified by the coordinates t in the Lie algebra. This also enables data sampling since we can attach prior distributions on the coordinates t so as to simulate a distribution on the group structure.

This Lie algebra parameterization without any constraints cannot guarantee a group structure to be decomposed into subgroups with each subgroup independently parameterized by a single coordinate (homomorphism between the group and the coordinate space), e.g. $\exp(t_1 A_1 + t_2 A_2) \neq \exp(t_1 A_1)\exp(t_2 A_2)$. This is not desirable since we would like a dimension in the coordinate t_i to identify a single subgroup $\exp(t_i A_i)$ and thus further represent a single variation in the data space. If this is not satisfied, the disentanglement is not achieved since the data variations are not encoded into separate subspaces $(t_1, t_2, ..., t_m)$. In the next two paragraphs we discuss two options to solve this problem.

One-parameter subgroup decomposition constraint. Based on our Lie algebra parameterization (Eq. 4), we have the following proposition to decompose a Lie group into one-parameter subgroups:

Proposition 1. *If $A_i A_j = A_j A_i, \forall i, j$, then*

$$\exp(t_1 A_1 + t_2 A_2 + ... t_m A_m)$$
$$= \exp(t_1 A_1)\exp(t_2 A_2)...\exp(t_m A_m) \quad (5)$$
$$= \prod_{\text{perm}(i)} \exp(t_i A_i). \quad (6)$$

Proof. See Appendix 1. □

Eq. 6 means the equation holds for any permutation of the index i. This decomposition ensures that the group structure is decomposed into subgroups with each one parameterized by a single coordinate t_i. Via this constraint, the data variations can be considered as disentangled into individual latent dimensions t_i's if an equivariance between the data variations and the group structure is learned. Since this decomposition holds for any permutation of the subgroups, the order of the subgroups would not influence the composition results thus the original group becomes commutative in terms of the subgroups. This shows a limitation of our method where it cannot disentangle variations that cannot be equivariantly represented by a commutative Lie group, e.g. 3D rotation decomposition along three orthogonal axes.

Hessian Penalty constraint. Besides learning disentanglement via enforcing the subgroup decomposition, we can also incorporate other useful disentanglement constraints like Hessian Penalty (Peebles et al., 2020) to the group representation. The Hessian Penalty assumes that the Hessian matrix with respect to a disentangled representation is always zero since the variation controlled by a dimension should not be a function of another dimension (independent):

$$H_{ij} = \frac{\partial^2 f(z)}{\partial z_i \partial z_j} = \frac{\partial}{\partial z_j}\Big(\frac{\partial f(z)}{\partial z_i}\Big) = 0, \quad (7)$$

where z is the disentangled representation, and $f(\cdot)$ is a function of z. Our Lie algebra parameterization is compatible with this constraint, and we have the following proposition:

Proposition 2. *If $A_i A_j = 0, \forall i \neq j$, then*

$$H_{ij} = \frac{\partial^2 g(t)}{\partial t_i \partial t_j} = 0, \quad (8)$$

where g is the map defined in Eq. 4.

Proof. See Appendix 2. □

Note that this is a more strict constraint than Proposition 1 since $A_i A_j = A_j A_i$ is implied by $A_i A_j = 0$, which also enforces the commutative group decomposition. This constraint further ensures that the dynamics caused by a subgroup is not affected by other subgroups at the group representation level (independent). Different from the original Hessian Penalty paper (Peebles et al., 2020) where the constraint is implemented on multiple feature maps with an unbiased stochastic approximator, our method penalizes the model only on the Lie group structure (using the Lie algebra basis), which is a different and a simpler implementation.

In summary, the Lie algebra parameterization enables practical learning by converting an optimization problem on groups to vector spaces. It also enables new constraints for encouraging disentanglement by enforcing commutative group decomposition and Hessian penalty.

4.3. Commutative Lie Group VAE

In this section, we present a simple model to learn disentangled representations using the group-related techniques proposed in the last section. We first introduce a VAE-variant (see Appendix 3 for an introduction of VAE) called bottleneck-VAE which forces a layer of feature in the encoder to match a layer of feature in the decoder. The model is a realization of the maximization of a lower bound of $\log p(x)$, which is presented in the following proposition:

Proposition 3. *Suppose two latent variables z and t are used to model the log-likelihood of data x, then we have:*

$$\log p(x) \geq \mathcal{L}_{\textit{bottleneck}}(x, z, t)$$
$$= \mathbb{E}_{q(z|x)}\mathbb{E}_{q(t|x,z)}\log p(x, z|t)$$
$$- \mathbb{E}_{q(z|x)}KL(q(t|x, z)||p(t)) - \mathbb{E}_{q(z|x)}\log q(z|x) \quad (9)$$
$$= \mathbb{E}_{q(z|x)q(t|z)}\log p(x|z)p(z|t)$$
$$- \mathbb{E}_{q(z|x)}KL(q(t|z)||p(t)) - \mathbb{E}_{q(z|x)}\log q(z|x), \quad (10)$$

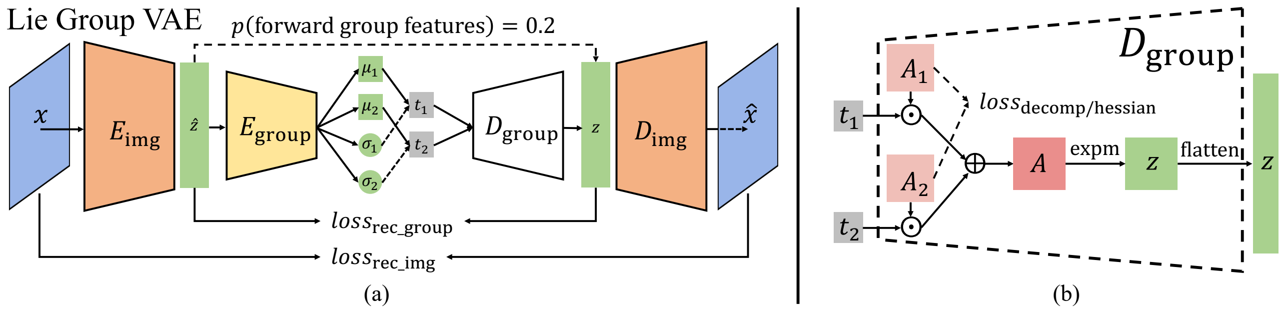

Figure 2. (a) The overall architecture of our proposed Lie Group VAE (bottleneck-VAE if D_{group} is a general layer; Commutative Lie Group VAE if constraints from Proposition 1 and 2 are applied). An input image is fed into the encoder E_{img} to obtain the encoded group representation $\hat{z}$. The representation fed through a group encoder E_{group} (a two-layer MLP) to obtain the Lie algebra coordinates t using reparameterization trick. The coordinates are then fed through a group decoder (exponential mapping layer) to obtain the group representation z, which is then fed through a decoder to reconstruct the image. (b) Detailed illustration of the exponential mapping layer D_{group}. For each coordinate scalar t_i we learn a Lie algebra basis A_i (a matrix), where constraints from Proposition 1 and 2 can be applied as regularizations. $\odot$ denotes broadcast multiplication, $\oplus$ sums up the matrices, and expm is the matrix exponential function. The obtained group representation is flattened to a vector before output.

where Eq. 10 holds because we assume Markov property: $q(t|z) = q(t|x, z)$*,* $p(x|z, t) = p(x|z)$*.*

Proof. See Appendix 4. □

The variable z is the feature to be shared in the encoder and the decoder. In practice, we model $p(z|t)$, $p(x|z)$, $q(z|x)$ to be deterministic networks (D_{group}, D_{img}, E_{img} in Fig. 2 (a)) while $q(t|z)$ to be a stochastic network (E_{group} in Fig. 2 (a)). The first term in Eq. 10 is implemented as two reconstruction losses on x and z respectively ($loss_{\text{rec\_img}}$ and $loss_{\text{rec\_group}}$ in Fig. 2 (a)), while the second term (KL-divergence) is implemented the same as in a standard VAE. The third term is the entropy of z when x is fixed, which is a constant since we model $q(z|x)$ to be a deterministic network, and is not implemented in practice. We forward the output feature of the encoder E_{img} directly to the decoder D_{img} at the probability of 0.2. This forces that both features (z and $\hat{z}$) are trained to reconstruct the same image so that they are encouraged to represent the same (high-level) information.

The Lie group structure is imposed on the (shared) feature z in the bottleneck-VAE, and it is easily implemented by constructing the network $p(z|t)$ (D_{group}) to be a Lie algebra parameterization $z = g(t)$ as defined in Eq. 4. In Fig. 2 (b) we show the details of how this module is implemented. For each input coordinate t_j we learn a Lie algebra basis element (a matrix) A_j in shape $|V| \times |V|$ (recall that V denotes the vector space on which the group is acting). The coordinates and basis are aggregated and fed into a matrix exponential mapping layer, resulting in a group representation. Popular deep learning toolkits TensorFlow (Abadi et al., 2015) and Pytorch (Paszke et al., 2019) both offer in-built differentiable implementations for matrix exponential map, which are computed by approximation methods proposed in (Higham, 2009; Bader et al., 2019). We name a bottleneck-VAE equipped with a Lie group structure as a Lie Group VAE.

Note that the bottleneck-VAE is essential to the realization of learning equivariance with a group structure because the encoder network needs to map the data directly onto a group structure so that it can be encouraged to learn the correspondance between the data variations and the group transformations. If there is no such a feature-sharing constraint (using plain VAE as a backbone), the encoder becomes a regular neural network and is not trying to learn a equivariance on the group structure but on the vector space, and the exponential mapping layer is just to assist the decoder to reconstruct the input data.

The one-parameter decomposition constraint and the Hessian penalty constraint can be directly applied as regularizations on Lie algebra basis $\{A_j\}_{j=1}^m$ (see Proposition 1 and 2). We name a Lie Group VAE equipped with either constraint a Commutative Lie Group VAE as both constraints enforce a commutative Lie group decomposition.

5. Experiments

We conduct experiments by following the general unsupervised disentanglement learning setup, i.e. training models on a dataset without any supervision and evaluate the quality of disentanglement by metrics on synthetic datasets and by latent traversal inspection on real-world datasets. Implementation details are shown in Appendix 5. Code is available at https://github.com/zhuxinqimac/CommutativeLieGroupVAE-Pytorch.

Models	FVM	SAP	MIG	DCI
VAE	$69.4_{\pm 10.9}$	$19.7_{\pm 10.6}$	$7.8_{\pm 6.4}$	$8.1_{\pm 4.1}$
+bottle	$74.6_{\pm 8.1}$	$29.2_{\pm 12.1}$	$12.9_{\pm 6.6}$	$11.6_{\pm 3.3}$
+exp	$\mathbf{83.6}_{\pm 3.2}$	$\mathbf{40.7}_{\pm 12.2}$	$\mathbf{17.2}_{\pm 6.8}$	$\mathbf{15.1}_{\pm 2.4}$

Table 1. Ablation study of bottleneck-VAE and exponential map on DSprites.

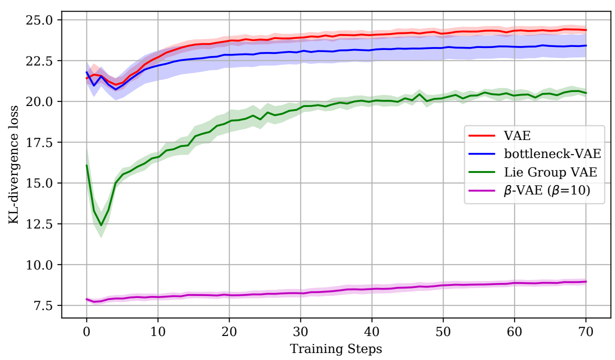

Figure 3. How the KL-divergence loss ($KL(q(t|x)||p(t))$) evolves during training for different models.

5.1. Synthetic datasets

We conduct experiments on the two most popular disentanglement datasets: DSprites (Matthey et al., 2017) and 3DShapes (Kim & Mnih, 2018). The DSprites dataset consists of 64×64 2D shapes rendered with 5 independent generative factors, i.e. *shape* (3 values), *scale* (6 values), *orientation* (40 values), *x position* (32 values), and *y position* (32 values). The total number of data samples is 737,280, with each factor combination appears exactly once. The 3DShapes dataset contains 64×64 images of 3D shapes generated from 6 independent factors, *floor color* (10 values), *wall color* (10 values), *object color* (10 values), *scale* (8 values), *shape* (4 values), *orientation* (15 values). There are 480,000 images in total in this dataset. For evaluation, we report results with various disentanglement metrics for an overall and robust evaluation. The used metrics include FactorVAE metric (FVM) (Kim & Mnih, 2018), SAP metric (Kumar et al., 2018), Mutual Information Gap (MIG) (Chen et al., 2018), and DCI Disentanglement metric (Eastwood & Williams, 2018). All the reported scores shown in this paper are averaged by 10 random runs. For ablation studies, we randomly split the dataset into training set (9/10) and test set (1/10) in each run, and compute evaluation scores on the test set. For the state-of-the-art comparison, we follow the tradition by training and evaluating on the whole dataset. We report the mean and standard deviation in all tables.

Effectiveness of the Proposed Components. We first investigate if the proposed model components can contribute to the disentanglement learning individually. In Table 1, we

$\text{Size}_{\text{group}}$	FVM	SAP	MIG	DCI
4	$23.6_{\pm 3.3}$	$6.3_{\pm 6.0}$	$4.2_{\pm 3.9}$	$3_{\pm 0.5}$
9	$57.4_{\pm 5.8}$	$34.1_{\pm 12.9}$	$17.3_{\pm 7.4}$	$12.4_{\pm 4.4}$
25	$79.8_{\pm 2.8}$	$39.6_{\pm 13.4}$	$20.6_{\pm 8.5}$	$19.9_{\pm 3.8}$
64	$82.7_{\pm 3.7}$	$42.2_{\pm 12.5}$	$22.1_{\pm 10.1}$	$\mathbf{20.0}_{\pm 6.8}$
81	$84.4_{\pm 2.6}$	$45.2_{\pm 10.5}$	$23.0_{\pm 8.4}$	$19.6_{\pm 6.3}$
100	$\mathbf{85.5}_{\pm 2.2}$	$\mathbf{50.8}_{\pm 5.0}$	$\mathbf{25.4}_{\pm 6.1}$	$19.7_{\pm 4.6}$

Table 2. Ablation study of group size on DSprites.

show how much a VAE can gain from these incorporated modules in terms of disentanglement scores. In Fig. 3 we show how the KL-divergence loss evolves. In the table, the +bottle entry corresponds to the introduced bottleneck-VAE which extends the plain VAE with an additional constraint by enforcing a shared layer of features between the encoder and decoder. Although its disentanglement results are still evidently inferior than other complete models (e.g. see results in other tables), it is quite surprising such a simple feature-sharing constraint can boost a VAE by an obvious margin. A potential explanation is that the bottleneck-VAE constrains the variations encoded in the latent codes since the model should *be very careful* to not change the to-be-shared feature too much or it becomes harder to reconstruct. This is beneficial to disentanglement since entanglement usually comes from codes which capture too much information so that they *overlap* or *intersect* with each other. This constraint on variation encoding can also be observed in Fig. 3 where for the +bottleneck (blue) line the KL-divergence between the posterior distribution and the latent prior becomes slightly smaller than a plain VAE (red). By adding the exponential mapping layer (D_{group}), which enforces the model to keep a group structure in the bottleneck, the disentanglement performance has been boosted to a competing level with other state-of-the-art methods. This is a key modification since from Fig. 3 we see the +exp (green) line evolves more elastically than the VAE baselines, e.g. VAE, +bottleneck, β-VAE (β=10). It shows that at the beginning the information encoded in the latent code is heavily constrained (moving closer towards the β-VAE) by the group-structure bottleneck. However because of the adaptivity of the group structure, more variations can be gradually learned as the training goes on. On the contrary, the KL losses shown in other VAE baselines change slowly, indicating the data variations are hard to be newly discovered during training in these models.

How the Group Representation Size Affects Disentanglement. In Table 2 we show how the disentanglement scores are affected by the choice of group representation size. Recall that the groups are represented by invertible matrices (subgroups of $GL(V)$) therefore they all have sizes of squared numbers. It can be expected that the larger the group representation is, the more likely the subgroups (in-

λ_{decomp}	FVM	SAP	MIG	DCI
0	$83.6_{\pm 3.2}$	$40.7_{\pm 12.2}$	$17.2_{\pm 6.8}$	$15.1_{\pm 2.4}$
5	$84.0_{\pm 3.9}$	$45.4_{\pm 11.5}$	$20.5_{\pm 6.9}$	$16.8_{\pm 4.3}$
20	$\mathbf{85.8}_{\pm 6.9}$	$48.7_{\pm 8.4}$	$23.6_{\pm 5.0}$	$18.2_{\pm 3.0}$
40	$85.5_{\pm 2.2}$	$\mathbf{50.8}_{\pm 5.0}$	$\mathbf{25.4}_{\pm 6.1}$	$\mathbf{19.7}_{\pm 4.6}$
80	$85.5_{\pm 4.8}$	$47.1_{\pm 8.6}$	$23.3_{\pm 6.2}$	$18.3_{\pm 6.5}$

Table 3. Ablation study of one-parameter decomposition on DSprites.

λ_{hessian}	FVM	SAP	MIG	DCI
0	$83.6_{\pm 3.2}$	$40.7_{\pm 12.2}$	$17.2_{\pm 6.8}$	$15.1_{\pm 2.4}$
5	$83.8_{\pm 2.4}$	$46.8_{\pm 12.8}$	$19.8_{\pm 8.6}$	$17.5_{\pm 5.6}$
20	$86.1_{\pm 1.8}$	$\mathbf{54.1}_{\pm 1.2}$	$\mathbf{29.7}_{\pm 3.1}$	$\mathbf{23.4}_{\pm 4.1}$
40	$\mathbf{86.2}_{\pm 1.8}$	$48.2_{\pm 1.9}$	$25.2_{\pm 8.4}$	$19.1_{\pm 4.1}$
80	$85.0_{\pm 1.6}$	$43.6_{\pm 11.3}$	$20.1_{\pm 8.4}$	$17.4_{\pm 4.2}$

Table 4. Ablation study of Hessian penalty on DSprites.

dexed by coordinates t_i's) can control different variations (due to the increased sparsity in a larger space). In the table, we can see the group representation of size 4 has almost no capability for disentanglement because a group of 2×2 matrices represents the linear transformations on a 2D plane, which is hard to be decomposed into more than two independent sub-transformations (e.g. scaling + rotation). When the representation size exceeds 25 (5×5), the model can easily find a decomposition to represent different data variations, and the disentanglement scores saturate. We use the group representation size 100 for all other experiments.

Effectiveness of One-parameter Subgroup Decomposition and Hessian Penalty Constraints. Now we investigate how the induced Lie-group related constraints benefit disentanglement learning. For each constraint we use a hyper-parameter $\lambda_{decomp}(\lambda_{hessian})$ to modulate the effect. The results of one-parameter subgroup decomposition is shown in Table 3, and Hessian Penalty is in Table 4. Both constraints can benefit disentanglement performance, where λ_{decomp} reaches the peak at 40 and $\lambda_{hessian}$ at 20. We can see the Hessian Penalty constraint is more effective than the subgroup decomposition as the performance gain in the former one is more significant (54.1 vs 50.8 on SAP and 29.7 vs 25.4 on MIG). This is due to that Hessian Penalty is a stronger constraint than the subgroup decomposition since it requires the Lie algebra basis elements to have mutual products of zeros while subgroup decomposition only requires their commutators to be zeros. This proves that forcing different subgroups to have independent effect on the final group representation is very beneficial to capture factorized data variations. In Fig. 4 we show the scatter plot of FactorVAE metric against reconstruction loss. We see our proposed Commutative Lie Group constraints boost the disentanglement performance at slight cost of reconstruction

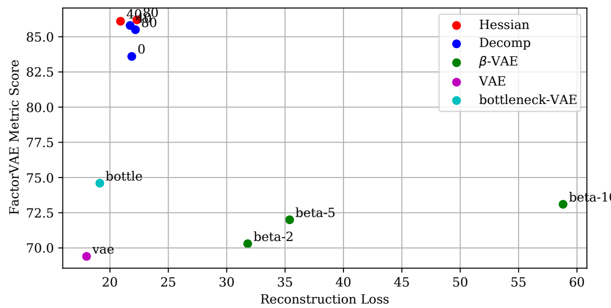

Figure 4. Reconstruction loss vs FactorVAE metric.

Model	DSprites	3DShapes
VAE	$69.4_{\pm 10.9}$	$83.6_{\pm 6.5}$
β-VAE	$74.4_{\pm 7.7}$	91 (Kim & Mnih, 2018)
Cascade-VAE	$81.74_{\pm 2.97}$	-
Factor-VAE	$82.15_{\pm 0.88}$	89 (Kim & Mnih, 2018)
Ours	$\mathbf{86.1}_{\pm 2.0}$	$\mathbf{93.2}_{\pm 4.0}$

Table 5. Unsupervised disentanglement state-of-the-art comparison on DSprites and 3DShapes.

quality, while the β-VAEs sacrifice reconstruction severely.

State-of-the-art Comparison. In Table 5 we compare our Commutative Lie Group VAEs with other state-of-the-art models for learning continuous latent variables on DSprites and 3DShapes datasets. We use the whole dataset to train and evaluate our models following the tradition of unsupervised disentanglement learning. Since most of the compared models only report the FVM score, we follow this standard and report the best model on FVM. We can see our Lie Group VAE model achieves the best performance among all compared models. It should be noticed that all other compared models enforce the statistical independence assumption to achieve disentanglement while our model is built based on another completely assumption by leveraging group structures. This indicates that our model has the potential to be further improved if independence assumption is concurrently enforced. Qualitative latent traversal results of our Commutative Lie Group VAE on both datasets are shown in Fig. 5.

5.2. Real-world Datasets

In this section we run our Commutative Lie Group VAE on real-world datasets including CelebA (Liu et al., 2014), Mnist (Lecun et al., 1998), and 3DChairs (Aubry et al., 2014).

CelebA dataset contains 202,599 images of cropped real-world human faces. We crop the center 128×128 area and

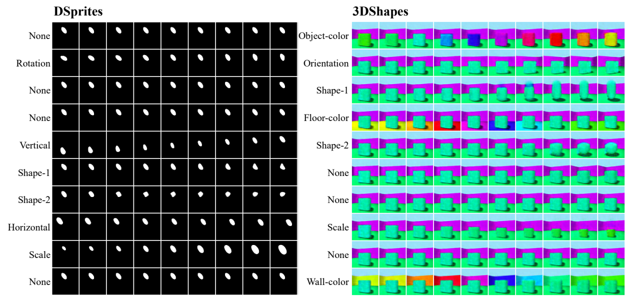

Figure 5. Latent traversals of our Commutative Lie Group VAE on DSprites and 3DShapes datasets.

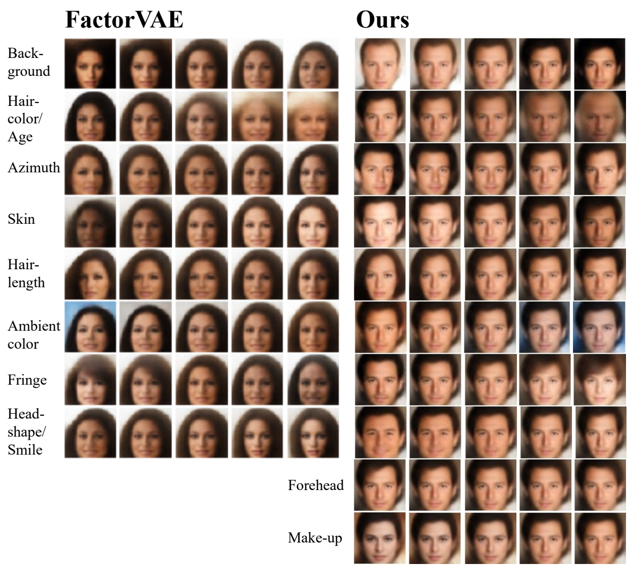

Figure 6. CelebA traversals compared with FactorVAE (Kim & Mnih, 2018).

resize the images to 64×64 for this experiment. In Fig. 6 we show the qualitative results of our model ($\lambda_{hessian} = 40$) trained on CelebA, and compare it with the FactorVAE baseline. For most of the attributes, our model can extract cleaner semantic variations than FactorVAE. For example, the *background* concept captured by FactorVAE is entangled with *smile* while in our model it is independently encoded. Additionally, our model learns to encode the semantics of *forehead hair-style* and *make-up* which are not shown in the FactorVAE.

Mnist dataset consists of handwritten digits (28×28 images) in 10 classes. We pad the images to size 32×32 for easier usage. We train our model with data of a same class to learn continuous variations. It is possible to integrate techniques for learning discrete latent variables (Dupont, 2018; Jeong & Song, 2019) in our Commutative Lie Group VAE for unsupervised classification, but we leave it for future work.

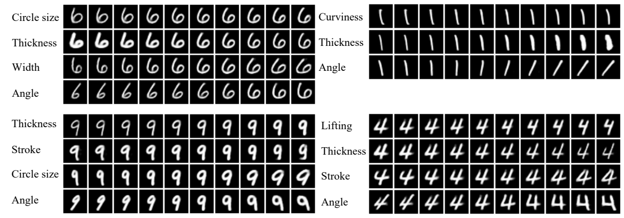

Figure 7. Per-class latent traversals on Mnist dataset.

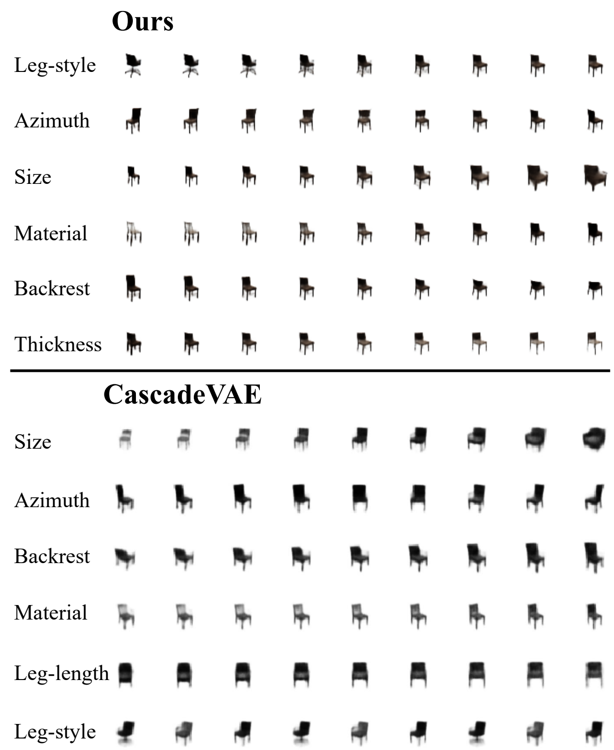

Figure 8. Comparing latent traversals of Commutative Lie Group VAE with CascadeVAE on 3DChairs.

In Fig. 7, we show some interesting semantic variations discovered by our model. We can see our model learns to control the circle size in *six* and *nine* with a variable, and also learns to represent the subtle variation of curviness in *one*. In *four*, our model discovers a *lifting* concept which controls the level of the horizontal line shown in 4's.

3DChairs dataset contains 86,366 RGB images of various chairs of resolution 64×64. In Fig. 8, we compare our model with the state-of-the-art CascadeVAE (Jeong & Song, 2019). Both models achieve similar disentanglement quality, showing that our model though based on a different assumption of group theory, can still achieve the same-level results as the state-of-the-art CascadeVAE based on information theory which models statistical independence.

6. Conclusion

In this paper we proposed to learn disentangled representations without supervision by capturing the data variations with an adaptive group structure. This is based on the idea that a group structure can represent data variations by group actions applied to itself. The advantage of using groups over usual vector spaces is that a group structure can not only equivariantly represent variations but also be adaptively optimized to fit the diversity in data variations. By replacing general vector representations with group representations, we can represent a data point with a group element on a group structure. To enable practical training, we adopted the Lie algebra parameterization and converted the learning problem on groups into the learning in linear spaces, which enables general optimization and sampling. Based on the parameterization, we introduced two simple commutative decomposition constraints for encouraging disentanglement, which are naturally derived from the one-parameter subgroup decomposition assumption and the Hessian Penalty assumption. To instantiate the group-based learning method, we introduced a variant of VAE called bottleneck-VAE, derived from a new lower bound of the data log-likelihood. We then proposed our (Commutative) Lie Group VAE by simply integrating an exponential mapping layer (with commutative decomposition constraints) into the decoder of the bottleneck-VAE. The proposed model achieved state-of-the-art performance in unsupervised disentanglement learning without adopting other regular constraints like statistical independence. Our proposed method is simple, elegant, and effective, and we believe this model exhibits a new direction for learning disentangled representations.

Acknowledgement

This work was supported in part by the Australian Research Council under Projects DE180101438 and DP210101859.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

Aubry, M., Maturana, D., Efros, A. A., Russell, B. C., and Sivic, J. Seeing 3d chairs: Exemplar part-based 2d-3d alignment using a large dataset of cad models. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3762–3769, 2014.

Bader, P., Blanes, S., and Casas, F. Computing the matrix exponential with an optimized taylor polynomial approximation. *Mathematics*, 7(12), 2019. ISSN 2227-7390. doi: 10.3390/math7121174.

Bekkers, E. J. B-spline cnns on lie groups. In *International Conference on Learning Representations*, 2020.

Bengio, Y., Courville, A. C., and Vincent, P. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35: 1798–1828, 2012.

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. Understanding disentangling in beta-vae. *ArXiv*, abs/1804.03599, 2018.

Caselles-Dupré, H., Garcia Ortiz, M., and Filliat, D. Symmetry-based disentangled representation learning requires interaction with environments. In *Advances in Neural Information Processing Systems*, pp. 4606–4615, 2019.

Chen, R. T. Q., Li, X., Grosse, R., and Duvenaud, D. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, 2018.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, 2016.

Cohen, T. and Welling, M. Learning the irreducible representations of commutative lie groups. In *ICML*, 2014.

Cohen, T. S. and Welling, M. Group equivariant convolutional networks. In *ICML*, 2016.

Cohen, T. S., Geiger, M., Köhler, J., and Welling, M. Spherical cnns. 2018.

Dieleman, S., De Fauw, J., and Kavukcuoglu, K. Exploiting cyclic symmetry in convolutional neural networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 18891898. JMLR.org, 2016.

Do, K. and Tran, T. Theory and evaluation metrics for learning disentangled representations. In *International Conference on Learning Representations*, 2020.

Dosovitskiy, A., Springenberg, J. T., and Brox, T. Learning to generate chairs with convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1538–1546, 2014.

Dupont, E. Learning disentangled joint continuous and discrete representations. In *NeurIPS*, 2018.

Eastwood, C. and Williams, C. K. I. A framework for the quantitative evaluation of disentangled representations. In *ICLR*, 2018.

Finzi, M., Stanton, S., Izmailov, P., and Wilson, A. G. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *ICML*, 2020.

Fuchs, F. B., Worrall, D. E., Fischer, V., and Welling, M. Se(3)-transformers: 3d roto-translation equivariant attention networks. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, 2020.

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

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. Towards a definition of disentangled representations, 2018.

Higham, N. J. The scaling and squaring method for the matrix exponential revisited. *SIAM Review*, 51(4):747–764, 2009. ISSN 00361445, 10957200.

Hinton, G. E., Krizhevsky, A., and Wang, S. D. Transforming auto-encoders. In *Proceedings of the 21th International Conference on Artificial Neural Networks - Volume Part I*, ICANN'11, pp. 4451, Berlin, Heidelberg, 2011. Springer-Verlag. ISBN 9783642217340.

Hoogeboom, E., Peters, J. W., Cohen, T. S., and Welling, M. Hexaconv. In *International Conference on Learning Representations*, 2018.

Jeon, I., Lee, W., and Kim, G. Ib-gan: Disentangled representation learning with information bottleneck gan. *ArXiv*, 2018.

Jeong, Y. and Song, H. O. Learning discrete and continuous factors of data via alternating disentanglement. In *ICML*, 2019.

Karras, T., Laine, S., and Aila, T. A style-based generator architecture for generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Kim, H. and Mnih, A. Disentangling by factorising. In *ICML*, 2018.

Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. Semi-supervised learning with deep generative models. In *NIPS*, 2014.

Kulkarni, T. D., Whitney, W. F., Kohli, P., and Tenenbaum, J. B. Deep convolutional inverse graphics network. In *NIPS*, 2015.

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

Lample, G., Zeghidour, N., Usunier, N., Bordes, A., Denoyer, L., and Ranzato, M. Fader networks: Manipulating images by sliding attributes. In *NIPS*, volume abs/1706.00409, 2017.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, 1989. doi: 10.1162/neco.1989.1.4.541.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

Li, Z., Murkute, J. V., Gyawali, P. K., and Wang, L.-W. Progressive learning and disentanglement of hierarchical representations. *ICLR*, abs/2002.10549, 2020.

Lin, Z., Thekumparampil, K. K., Fanti, G., and Oh, S. Infogan-cr: Disentangling generative adversarial networks with contrastive regularizers. *ICML*, 2020.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3730–3738, 2014.

Matthey, L., Higgins, I., Hassabis, D., and Lerchner, A. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Painter, M., Hare, J., and Prugel-Bennett, A. Linear disentangled representations and unsupervised action estimation, 2020.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Peebles, W., Peebles, J., Zhu, J.-Y., Efros, A. A., and Torralba, A. The hessian penalty: A weak prior for unsupervised disentanglement. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

Quessard, R., Barrett, T., and Clements, W. R. Learning group structure and disentangled representations of dynamical environments. *NIPS*, abs/2002.06991, 2020.

Reed, S. E., Sohn, K., Zhang, Y., and Lee, H. Learning to disentangle factors of variation with manifold interaction. In *ICML*, 2014.

Sosnovik, I., Szmaja, M., and Smeulders, A. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2020.

Worrall, D. and Brostow, G. Cubenet: Equivariance to 3d rotation and translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Worrall, D. and Welling, M. Deep scale-spaces: Equivariance over scale. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 7366–7378. Curran Associates, Inc., 2019.

Worrall, D. E., Garbin, S. J., Turmukhambetov, D., and Brostow, G. J. Harmonic networks: Deep translation and rotation equivariance. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7168–7177, 2017. doi: 10.1109/CVPR.2017.758.

Yan, X., Yang, J., Sohn, K., and Lee, H. Attribute2image: Conditional image generation from visual attributes. In *ECCV*, 2016.

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

Zhu, X., Xu, C., and Tao, D. Learning disentangled representations with latent variation predictability. In *ECCV*, 2020.

Zhu, X., Xu, C., and Tao, D. Where and what? examining interpretable disentangled representations. In *CVPR*, 2021.