

Group-based Learning of Disentangled Representations with Generalizability for Novel Contents

Haruo Hosoya

ATR International, Kyoto, Japan

hosoya@atr.jp

Abstract

Sensory data are often comprised of independent content and transformation factors. For example, face images may have shapes as content and poses as transformation. To infer separately these factors from given data, various “disentangling” models have been proposed. However, many of these are supervised or semi-supervised, either requiring attribute labels that are often unavailable or disallowing for generalization over new contents. In this study, we introduce a novel deep generative model, called group-based variational autoencoders. In this, we assume no explicit labels, but a weaker form of structure that groups together data instances having the same content but transformed differently; we thereby separately estimate a group-common factor as content and an instance-specific factor as transformation. This approach allows for learning to represent a general continuous space of contents, which can accommodate unseen contents. Despite the simplicity, our model succeeded in learning, from five datasets, content representations that are highly separate from the transformation representation and generalizable to data with novel contents. We further provide detailed analysis of the latent content code and show insight into how our model obtains the notable transformation invariance and content generalizability.

1 Introduction

Sensory data are often composed of multiple independent factors. For example, 3D face images usually consist of the content, representing how the face is shaped, and the transformation, representing how the face is posed or expressing. Humans are generally good at inferring and separating such factors from given inputs without much supervision. Moreover, once discovering such factors, one can automatically generalize this knowledge for novel contents, e.g., faces of unseen identities. How can such ability be achieved computationally?

The content-transformation separation problem [Tenenbaum and Freeman, 2000] has recently been attracting much

attention, in accordance with the progress of deep generative techniques such as variational autoencoders (VAE) [Kingma and Welling, 2014] and generative adversarial learning (GAN) [Goodfellow *et al.*, 2014], and nowadays often called “disentangling.” For this purpose, a number of supervised or semi-supervised models have been proposed exploiting labels in various ways. The most typical approach learns a generative model by explicitly supplying class labels to the content variable while extracting the remaining factor in the transformation variable [Kingma *et al.*, 2014; Cheung *et al.*, 2014; Siddharth *et al.*, 2017]. However, since the content representation in such approach is often restricted to the classes seen during training, it does not generalize for unseen classes. Other approaches use a more sophisticated method, such as adversarial learning, that exploits labels so as to make the content representation as irrelevant as possible to transformation [Wang *et al.*, 2017; Lample *et al.*, 2017; Mathieu *et al.*, 2016]. Although these approaches can potentially allow for generalization over new contents, their requirements of specific kinds of label are often difficult to fulfill, e.g., attribute labels corresponding to transformation.

In this study, we propose a new deep generative model, called group-based variational autoencoders (GVAE), for content-transformation separation. In GVAE, we do not require explicit labels in the dataset. Instead, we assume a weaker structure that groups together multiple data instances (like images) containing the same content but transformed differently. Then, our model learns to extract the content as the factor common within each group and the transformation as the factor specific to each instance. For example, from the set of groups of face images as in Figure 1A, the facial shape can be extracted as the group-common factor and the facial pose and expression as the instance-specific factor. Our grouping assumption is quite general in the sense that such structure can be derived not only from class-labeled data by grouping instances of the same class, but also from video data by grouping frames in the same or nearby sequence. Further, this approach allows for generalization over novel contents: we use a continuous variable to form a general “space” of content (e.g., a space of facial shapes) and thereby can represent an infinite number of contents including those not seen during training (Figure 1B).

Our specific contributions are (i) to describe a novel learning method that takes a grouped dataset (with no other labels)

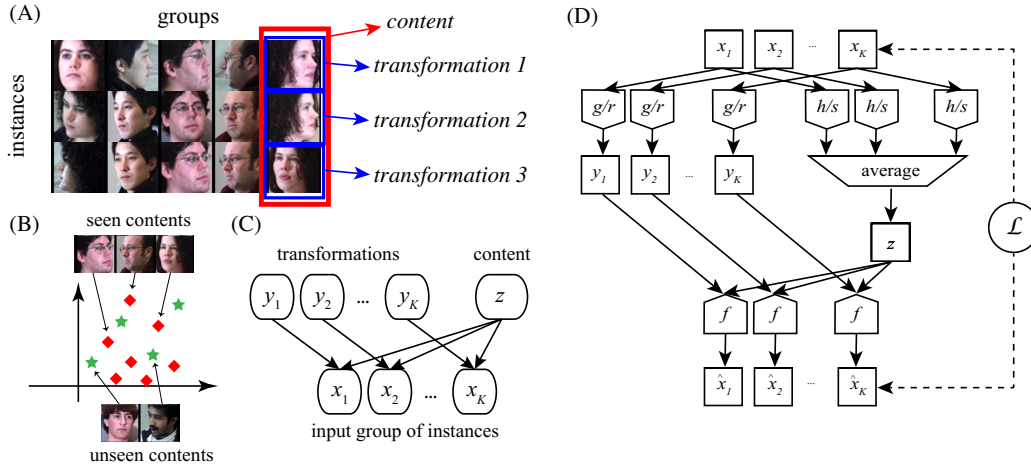


Figure 1: (A) Example of grouped data. Each column is a group of face images of the same person (content) with different views (transformation). (B) A continuous space of contents learned from training data can accommodate novel contents. (C) The graphical model. Each instance x_k is generated from the instance-specific transformation variable y_k and the group-common content variable z . (D) Algorithmic outline of GVAE learning. The encoders g and r given an individual image x_k compute the corresponding transformation representation y_k , while the encoders h and s followed by averaging compute the common content representation z . The decoder f then reconstructs each image \hat{x}_k by combining the corresponding transformation y_k and the common content z . (See text for details.)

and separately extracts group-common and instance-specific factors, (ii) to show, qualitatively and quantitatively, that this method learns highly separate representation of content and transformation and generalizes it to new contents, for five datasets, and (iii) to provide detailed analysis of latent code to reveal insight into the generalizable disentangling.

2 Related Work

As already mentioned, a number of supervised or semi-supervised approaches have been developed for disentangling. The simplest approach is to explicitly supply the content information and infer the transformation variable as the remaining factor [Kingma *et al.*, 2014; Cheung *et al.*, 2014; Siddharth *et al.*, 2017]. However, the prior studies in this approach have typically supplied class labels for supervision (since these are the only labels available in most datasets), represented them in categorical variables, and estimated a generative model conditioned on the class. Since such estimated model would have little useful information for unseen classes, generalization for new contents is difficult.

In other supervised approaches, more sophisticated mechanisms have been devised to exploit labels for discovering a transformation-invariant content representation in a continuous variable. For example, [Wang *et al.*, 2017] has introduced a mapping (deep net) from labels to latent code while maintaining consistency with another mapping from input images to the same latent code; [Lample *et al.*, 2017; Mathieu *et al.*, 2016] have used adversarial learning to make the content code as irrelevant as possible to transformation using labels. While effective disentangling can thus be achieved with the aid of labels, requirement for labels, in particular, attribute labels corresponding to transformation, seems rather restrictive.

Concurrently with ours, one recent study has developed another group-based method called Multi-Level VAE (MLVAE) [Bouchacourt *et al.*, 2018]. In this, they adopt a sophisticated technique called “evidence accumulation” for estimating the group-common factor. However, as we show later, this particular technique has an unfortunate property that the learned content representation often becomes dependent on the transformation, which potentially conflicts the goal of disentangling (Section 5). Indeed, due to this, MLVAE often results in quantitatively poorer disentangling representations compared to our GVAE (Section 4). In [Chen *et al.*, 2018], a group-based GAN method has been proposed for learning separate representation of content and transformation. However, their goal is rather different from ours, as they focus on randomly generating new realistic images while lacking a way to infer content and transformation information from given images.

Some studies have investigated disentangling for sequential data like video [Yang *et al.*, 2015; Denton and Birodkar, 2017]. They crucially assumed that the content variable remains similar in consecutive frames, as inspired by the temporal coherence principle [Földiák, 1991]. Although this idea is related to the group-based approach, the sequence-based methods have used additional mechanisms to exploit ordering among data instances, e.g., a recurrent neural network to predict future states from past states [Yang *et al.*, 2015] or adversarial training to take temporal structure into account [Denton and Birodkar, 2017].

Some unsupervised approaches require no label or grouping in data, but optimizes the efficiency of the latent representation, either by maximizing mutual information between hidden variables in a GAN-based model [Chen *et al.*, 2016] or by adjusting a so-called β -term in a variational lower bound [Higgins *et al.*, 2016]. Neither study reported generalizability over novel contents.

3 Methods

3.1 Model

We assume a dataset \mathcal{D} consisting of N groups, in which each group has K data instances, $(x_1^{(n)}, \dots, x_K^{(n)})$, where $x_k^{(n)} \in \mathcal{R}^D$ is a data instance indexed by the group number n and the instance number k . For example, Figure 1A shows a set of 5 groups of 3 data instances, where each instance is an image. (For brevity, we sometimes elide the superscript (n) below.) We assume independence between groups but not instances within a group. We intend that each group contains the same content with (possibly) different transformations. In other words, the factor common among the instances correspond to the content, while the factor differentiating them correspond to the transformation. (We do not require alignment in transformation at each instance number. Also, we do not forbid that different groups happen to contain the same content.)

For such grouped data, we consider the following probabilistic generative model with two types of hidden variables: the (instance-specific) transformation variables $y_1, \dots, y_K \in \mathcal{R}^L$ and the (group-common) content variable $z \in \mathcal{R}^M$ (Figure 1C):

$$p(z) = \mathcal{N}(0, I) \quad (1)$$

$$p(y_k) = \mathcal{N}(0, I) \quad (2)$$

$$p_\theta(x_k|y_k, z) = \mathcal{N}(f_\theta(y_k, z), I) \quad (3)$$

for $k = 1, \dots, K$. Here, f_θ is a decoder deep net with weight parameters θ . In the model, the content z or each transformation y_k is first generated from the standard Gaussian prior. Then, each instance x_k is generated from the decoder f_θ applied to the corresponding transformation y_k and the common content z , added with Gaussian noise of unit variance.

3.2 GVAE

For learning, we extend the VAE approach introduced by [Kingma and Welling, 2014], which uses encoder models based on deep nets to approximate posterior distributions for hidden variables. (Figure 1D illustrates the outline of the learning algorithm.) First, we estimate each transformation y_k from the corresponding input instance x_k as follows:

$$q_{\phi, \xi}(y_k|x_k) = \mathcal{N}(g_\phi(x_k), r_\xi(x_k)) \quad (4)$$

where we use an encoder deep net g_ϕ with weight parameters ϕ for estimating the mean and another deep net r_ξ (positively valued) with weight parameters ξ for estimating the dimension-wise variances.¹ For inference of content, we could likewise assume a pair of deep nets to estimate the content z from all instances x_1, \dots, x_K , but it cannot exploit symmetry in the instances in the same group. Instead, we take the following simpler approach:

$$q_{\psi, \pi}(z|x_1, \dots, x_K) = \mathcal{N}\left(\frac{1}{K} \sum_{k=1}^K h_\psi(x_k), \frac{1}{K} \sum_{k=1}^K s_\pi(x_k)\right) \quad (5)$$

¹Since we consider only Gaussians with diagonal covariances, we specify a vector of variances in the second parameter to Gaussian distribution as convention.

That is, the encoder deep nets h_ψ and s_π first each estimate the mean and variance of the posterior distribution of the individual content for each instance x_k . Then, we infer the common content z for all instances as the average of all the individual contents. Note that, therefore, the variance of z becomes the average of the individual variances. The intention here is that, as we attempt to reconstruct each instance x_k by the common content z with the individual transformation y_k , all the individual contents $h_\psi(x_1), \dots, h_\psi(x_K)$ are encouraged to converge to an equal value in the course of learning. Thus, z will eventually become a common factor of all x_k , while each y_k will become a differentiating factor.

To train the model, we consider the following variational lower bound of the log likelihood:

$$\begin{aligned} \log p_\theta(\mathbf{x}) &\geq \mathcal{E}_{q_{\phi, \xi, \psi, \pi}(\mathbf{y}, z|\mathbf{x})} \left[\sum_{k=1}^K \log p_\theta(x_k|y_k, z) \right] \\ &\quad - \sum_{k=1}^K \text{KL}(q_{\phi, \xi}(y_k|x_k) \parallel p(y_k)) \\ &\quad - \text{KL}(q_{\psi, \pi}(z|\mathbf{x}) \parallel p(z)) = \mathcal{L}(\mathbf{x}) \end{aligned} \quad (6)$$

where $\mathbf{x} = (x_1, \dots, x_K)$ and $\mathbf{y} = (y_1, \dots, y_K)$. Then, our goal is to maximize the lower bound for the whole dataset \mathcal{D} :

$$\mathcal{L}(\mathcal{D}) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(\mathbf{x}^{(n)}) \quad (7)$$

This can be solved by the standard VAE techniques (stochastic variational Bayes, reparametrization, etc.; [Kingma and Welling, 2014]).

3.3 Experimental Set-up

We prepared the following five datasets. (1) Multi-PIE: multi-viewed natural face images derived from [Gross *et al.*, 2010]; grouping by the subject and (cloth/hair) fashion while varying the view and expression; the training and test sets with disjoint 268 and 69 subjects. (2) Chairs: multi-viewed synthetic chair images derived from [Dosovitskiy and Springenberg, 2015]; grouping by the chairs type while varying the view; the training and test sets with disjoint 650 and 159 types. (3) KTH: image frames from video clips of human (only pedestrian) motion derived from [Schüldt *et al.*, 2004]; grouping by the video clip while varying the position of the subject; the training and test sets with disjoint 20 and 5 subjects. (4) Sprites: multi-posed synthetic game character images [Reed *et al.*, 2015]; grouping by the character id while varying the pose and wear; the training and test sets with disjoint 500 and 100 ids. (5) NORB: multi-viewed toy images [LeCun *et al.*, 2004]; grouping by the toy id varying the view and lighting; the training and test sets with disjoint 25 and 25 ids. Note that labels were used only to form groups for some datasets but never explicitly given to the training algorithm; no label was used to form groups in KTH (video data). See Appendix A for details of these datasets.

For each dataset, we built a number of GVAE models with convolutional neural nets for the encoders (g , r , h , and s) and a deconvolutional neural net as the decoder (f). Each encoder



Figure 2: Illustration of generalizable disentangling in a GVAE model trained with Multi-PIE. (A) Swapping for training images. Each image in the matrix was generated from the content representation of a sample image in the top row and the transformation representation of another sample image in the left-most column. Red box: sample image. (B) Interpolation for training images. For two training sample images, each image was generated from a linear interpolation (or extrapolation) of the content representations of the two sample images, in conjunction with the transformation representation of either sample image. Green box: images corresponding to the sample. (C,D) Analogous swapping and interpolation for test images. The test images have distinct facial identities from the training images.

had three convolution layers followed by two fully connected layers (intervened with RELU nonlinearity); each decoder had a reverse architecture. The top layer of encoder r or s had nonlinearity to ensure positivity: $F(a) = \exp(a/2)$. We used the same architecture for all models with the transformation dimension $L = 3$ (except $L = 2$ for Chairs and KTH) and the content dimension $M = 100$. See Appendix B for details of the architecture.

We found it crucial to choose a very low dimension for the transformation variable ($L = 2$ or 3). In this way, the transformation variable was endowed with a just enough space to encode the instance-specific factor and no spare space to encode the common factor. If the dimensionality was instead much raised, then the transformation variable would learn to represent all aspects of inputs including the contents and thereby the content variable would become degenerate. Apart from this, the results were generally stable across choices of other parameters such as deep net architectures.

To train each model, we first randomly initialized the weight parameters of the encoders and decoder and then optimized the objective (7) with respect to the weight parameters using the training set. Training proceeded by mini-batches (size 100), where each group was formed on the fly by randomly choosing 5 images according to the dataset-specific grouping policy described above ($K = 5$). We used Adam optimizer [Kingma and Ba, 2015] with the recommended optimization parameters. During training, we monitored effectiveness of each latent dimension (s.d. ≥ 0.1) and discarded models with very low effective dimensions (0 for y , or ≤ 3 for z).

4 Results

4.1 Examples

We first illustrate the learned disentangled representations in two ways. *Swapping*: given two lists of inputs x_1, \dots, x_I and x'_1, \dots, x'_J , we show a matrix of images, each generated from the content representation estimated from an input x_i and the transformation representation estimated from another input x'_j : $f(g(x'_j), h(x_i))$. *Interpolation*: given two inputs x_1 and x_2 , we show a matrix of images, each generated from a linear interpolation of the content representations estimated for both inputs, with the transformation representation for either input: $f(g(x_i), (1 - \alpha)h(x_1) + \alpha h(x_2))$ for $0 \leq \alpha \leq 1$ for $i = 1, 2$; extrapolation is also possible with α outside this range. Note that we estimated the latent representation from a single image (with no grouping) and did not use the variance encoders r and s .

Figure 2A shows a swapping matrix for a GVAE model trained with Multi-PIE dataset. The top row and the left-most column show two lists of sample images in the training set. In the matrix, we can observe reasonably successful disentangling: each generated image reflects the subject of the corresponding input in the first list and the view of the corresponding input in the second list. Figure 2B shows an interpolation matrix from the same model for sample training images. Note the smooth transition of the generated images from one identity to another. Also some generated images for extrapolated contents exhibit exaggerated facial features (e.g., hair).

success rate (%)	1-shot classification			5-shot classification			chance
	GVAE	MLVAE	VAE	GVAE	MLVAE	VAE	
MultiPIE	44.3 ±3.2	24.5±3.9	9.1±0.8	64.0 ±2.0	48.5±4.3	19.0±1.6	0.3
Chairs	58.4 ±6.7	53.3±5.2	18.4±3.0	82.7 ±3.9	80.7±4.3	41.3±6.3	0.6
KTH	27.1±1.6	30.1 ±0.6	14.2±1.7	48.9±1.8	55.3 ±0.8	35.6±3.1	0.4
Sprites	81.3 ±7.4	53.5±16.7	5.0±1.4	84.7 ±4.8	65.0±14.6	9.5±3.1	1.0
NORB	37.5 ±1.3	31.2±1.9	12.9±3.7	44.7 ±2.1	42.7±2.9	21.8±7.0	4.0

Table 1: Quantitative comparison of generalizability of disentangled representations evaluated by few-shot classification. The success rate (mean and s.d.) is shown for each method and for each dataset in the 1-shot or 5-shot case, along with the chance level.

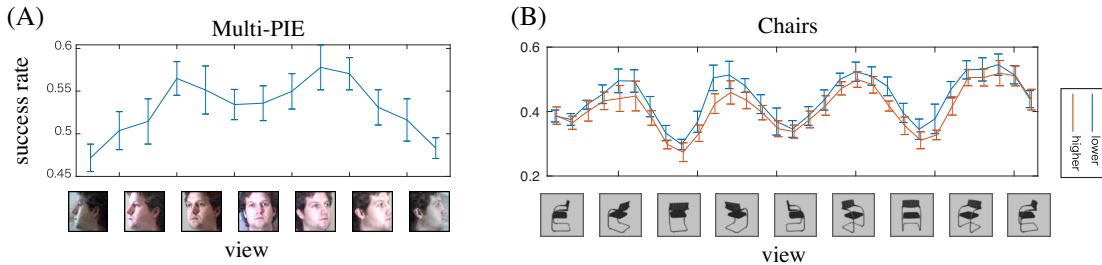


Figure 3: (A) The success rate (y-axis) for each view (x-axis; two down-looking views are omitted) from a GVAE model trained with Multi-PIE. (B) Analogous result for Chairs. (Two colors correspond to different vertical angles of view; see legend).

Figure 2CD shows swapping and interpolation matrices for sample images in the test set. Generalization of disentangling to the test case can be observed with a quality more or less similar to the training case. This is quite remarkable, given the fact that none of the subjects here had been seen during the training.

4.2 Quantitative Comparison

We next show the results of our quantitative evaluation. We here use accuracy of few-shot classification as criterion. The rationale is that, since the learned content representation is expected to eliminate information on transformation, this should allow for transformation-invariant recognition of objects. In particular, if the content representation is highly independent from transformation and highly generalizable for novel contents, then it should ideally be sufficient to hold the content representation of a single example of each unseen class in order to classify the remaining images (one-shot classification). Thus, the better the generalizable disentangled representation is, the more accurate the few-shot classification should be.

Our evaluation procedure goes as follows. For a dataset, we formed a split of the test images into the gallery including S random images for each class and the probe including the remaining images (S -shot case). Here, a class referred to each subject/fashion combination in Multi-PIE (378 test classes), to each chair type in Chairs (159 test classes), to each video clip id in KTH (240 test classes), to each sprite id in Sprites (100 test classes), and to each toy id in NORB (25 test classes). Then, for a trained model, we classified each probe image as the class of the gallery image that had the maximal cosine similarity with the input probe image in the space of content variable z ; classification succeeded if the inferred class matched the actual class; the success rate was averaged

over 100 different splits. We examined 10 separately trained models (different only in the initial weights) for each dataset.

For comparison, we also trained models with the MLVAE method [Bouchacourt *et al.*, 2018], another group-based method; the training condition was the same. In addition, we trained GVAE models with no grouping ($K = 1$). This in fact corresponds to a basic VAE model with a single variable since the content and transformation variables, without grouping, can be integrated without loss of generality.

Table 1 summarizes the result, showing the success rate (mean and s.d. over the 10 model instances) of one-shot or five-shot classification for each method and for each dataset. In most cases, GVAE outperformed the other methods. In particular, for Multi-PIE and Sprites datasets, the scores for GVAE were much higher than MLVAE. VAE models always performed poorly (though far better than the chance level); this condition in fact failed disentangling with no clear content-transformation separation and often corrupted generated images. Thus, grouping was indeed crucial.

As a side interest, we wondered which view of 3D objects led to more successful recognition. Figure 3 shows the view-wise success rates of one-shot classification in GVAE models for Multi-PIE and Chairs. We found that, in both cases, diagonal views always gave better success rates than profile or frontal views. This result is intuitive since we can perceive better the entire shape of a face or chair from diagonal views than other views. Perhaps, this is the reason why photos of actors or furniture items are typically taken in diagonal views.

5 Analysis of Content Coding

Why did GVAE achieve better generalizable disentangling than MLVAE? To gain insight into this question, we have conducted detailed analysis of content representation. First

of all, the most essential difference of MLVAE [Bouchacourt *et al.*, 2018] from GVAE is the way inferring the content (equation 5), where the posterior distribution is estimated as the product of Gaussians for individual instances (“evidence accumulation”):

$$q_\psi(z|x_1, \dots, x_K) = \frac{1}{Z} \prod_{k=1}^K \mathcal{N}(h_\psi(x_k), s_\pi(x_k)) \quad (8)$$

where Z is the normalizing constant. Since a product of Gaussians is a Gaussian, the above definition can be rewritten as:

$$(8) = \mathcal{N}\left(\frac{\sum_k h_\psi(x_k)/s_\pi(x_k)}{\sum_k 1/s_\pi(x_k)}, \frac{1}{\sum_k 1/s_\pi(x_k)}\right) \quad (9)$$

where all the multiplications and divisions are dimension-wise. Note that the mean of the Gaussian has the form of weighted average where the weights are the precisions $1/s_\pi(x_k)$. Although GVAE uses simple averaging (equation 5), it is not a special case of MLVAE since the variance has a very different form. We claim that this evidence accumulation technique unfortunately leads to view-dependent content representation in MLVAE unlike GVAE.

First of all, we found that, generally, MLVAE models had ~ 3 times larger number of effective content dimensions than GVAE models. To understand why, we picked up one example MLVAE model for Multi-PIE, which had 40 effective content dimensions, and inspected the structure of the content representation in the 8 most effective dimensions (i.e., with the largest s.d.’s). In Figure 4A, each scatter plot (blue) shows the estimated precisions $1/s(x)$ for each view of test images (left-profile, frontal, right-profile, etc.), separately for each content dimension. The estimated precisions tended to be peaked at a particular view while they went down to very low values elsewhere (e.g., dim #1 had peak at 45° left profile), and the peak view was different from dimension to dimension. This was starkly different in a GVAE model (which had 13 effective content dimensions). Figure 4B shows the corresponding plot for this model, where the estimated precisions were only slightly higher for frontal views compared to profile views for most dimensions. Recall that, in MLVAE, because of the weighted-average form (equation 9), each estimated mean is multiplied with the corresponding estimated precision. Since the estimated precisions were very low for the non-peak views (Figure 4A), the content dimension would have strong “impact” only around the peak views.

To confirm this, we quantified how much each content dimension had impact on image generation. We first estimated the latent variables $y = g(x)$ and $z = h(x)$ for each test image x and then generated two new images using the estimated variables but modifying d -th content dimension. Thereafter, we calculated the (normalized) Euclidean distance between the two generated images:

$$I_d(x) = \frac{\|f(y, z_d^-) - f(y, z_d^+)\|}{(\|f(y, z_d^-)\| + \|f(y, z_d^+)\|)/2}$$

where

$$z_d^- = (z_1, \dots, z_{d-1}, \mu_d - 3\sigma_d, z_{d+1}, \dots, z_M)$$

$$z_d^+ = (z_1, \dots, z_{d-1}, \mu_d + 3\sigma_d, z_{d+1}, \dots, z_M)$$

using the mean μ_d and s.d. σ_d of z_d (over the test data). We called $I_d(x)$ “generative impact index.” If the two generated images are very different, then the content dimension has a large index; otherwise, it has a small index. In Figure 4AB, each curve (red) shows generative impact indices for each view at each content dimension. We can see that the generative impact consistently followed the magnitudes of precisions in both MLVAE and GVAE. As a consequence, in MLVAE, each dimension had much more impact on image generation for high-precision views than low-precision views, whereas such contrast was more moderate in GVAE (Figure 4D). Also, note that the peak views covered a broad range of views in MLVAE, while they were concentrated around the frontal view (Figure 4C).

Thus, MLVAE had view-dependent coding of content. This in fact explains the generally larger number of effective content dimensionality in MLVAE. That is, in MLVAE, different content dimensions seemed to be used for representing different views and therefore encoding all views required a larger number of dimensions, whereas each dimension seemed to play an equal role in all views in GVAE.

The view dependency of the estimated precisions in MLVAE may be related to uncertainty coding of facial features. That is, since a single image provides only partial information for the content, the inference necessarily includes ambiguity. For example, from a frontal face image, we are sure about the eyes, nose, and mouth, but less sure about the ears; from a profile face image, we are sure about the visible side of the face, but much less sure about the invisible side. Thus, by taking uncertainty into account, we might be able to infer a more accurate content representation when integrating estimated content information from different views. However, the view-dependent representation seems to go in the opposite direction to the goal of disentangling—to discover view-invariant representation. This explains the reason for the observed lower performance of MLVAE in few-shot classification.

6 Conclusion

We have proposed group-based VAE as a novel method for learning disentangled representations from grouped data. Despite the simplicity, our method achieved effective disentangling of content and transformation that generalized for test images with new contents. We quantitatively evaluated generalization performance using few-shot classification and showed superiority of our approach to MLVAE for most datasets. Our detailed analysis revealed that the performance difference was due to the evidence accumulation technique used in MLVAE causing transformation dependency of content representation.

Future investigation may pursue improvement of disentangling by incorporating some additional constraints such as adversarial learning. Also, disentangling for more realistic and irregular dataset would be important. Lastly, as the present model was initially inspired by findings in visual neuroscience and started as a continuation of our previous theoretical study on the primate visual system [Hosoya and Hyvärinen, 2016; Hosoya and Hyvärinen, 2017], we are keen to investigate how our model here could serve to explain the underlying learning principle in the higher vision.

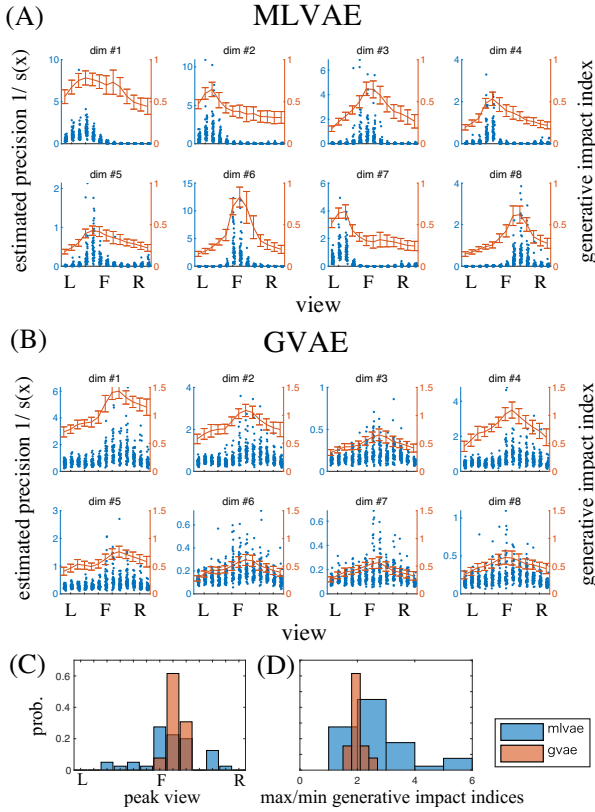


Figure 4: Analysis of latent content representations. (A) Blue: the distributions of estimated precisions $1/s(x)$ for each view for each of top 8 content dimensions in an MLVAE model (scale: $\times 10^3$). Red: the view-wise normalized distances between two images generated from latent variables differing in a particular content dimension (generative impact index). (B) Analogous results for a GVAE model. (C) The distribution of peak views (in terms of generative impact indices) for the effective dimensions in the MLVAE or GVAE model (legend). (D) The distribution of ratios of maximum and minimum generative impact indices in each model.

Acknowledgments

The author expresses special thanks to Aapo Hyvärinen, Mohammad Emtiyaz Khan, and Motoaki Kawanabe for precious comments and suggestions. This work has been supported by the Commissioned Research of National Institute of Information and Communications Technology (1940201), the New Energy and Industrial Technology Development Organization (P15009), and Grants-in-Aid for Scientific Research (18H05021, 18K11517, and 19H04999).

A Dataset Details

Multi-PIE The original dataset [Gross *et al.*, 2010] consists of natural face images of 337 subjects in 15 views, 3 expressions, and 4 (cloth/hair) fashions. We used only images under a medium lighting (condition 9) and cropped and resized them using the manual landmark annotations in [El Shafey *et al.*, 2013] ($64 \times 64 \times 3$ pixels). We split the data into a training set with ~ 32 K images of 268 subjects and a test set with ~ 6 K

images of the remaining 69 subjects. We formed each group by randomly choosing face images of the same subject and fashion, but varying the view and the expression.

Chairs The original dataset [Dosovitskiy and Springenberg, 2015] consists of synthetic multi-viewed chair images of various types rendered from 3D models. We used a convenient subset [Yang *et al.*, 2015] including images ($64 \times 64 \times 3$ pixels) of 809 chair types in 62 views (31 horizontal and 2 vertical angles). We split the data into a training set with ~ 40 K images of 650 chair types and a test set with ~ 10 K images of the remaining 159 chair types. We formed each group by randomly selecting chair images of the same type but varying the view.

KTH The original dataset [Schüldt *et al.*, 2004] consists of video clips of human motion of 6 types by 25 subjects, where each clip shows one type of motion by a single subject with varied settings (background, motion direction, cloth, camera-work, etc.). From a subset with pedestrian motions (walking, running, and jogging), we created a training set with image frames (64×64 pixels) from 960 video clips of 20 subjects and a test set with image frames from 240 video clips of the remaining 5 subjects; we removed frames showing no person. We formed each group by randomly selecting image frames in the same video clip.

Sprites The original dataset [Reed *et al.*, 2015] consists of synthetic game character images. A structured label is given to each image to specify the features of the character; the images are varied for the pose and wear. We assigned a unique id to each different structured label. We split the data into a training set with ~ 89 K images of 500 ids and a test set with ~ 19 K images of 100 other ids. We formed each group by randomly selecting images of the same id but varying the pose and wear.

NORB The original dataset [LeCun *et al.*, 2004] consists of images of 50 toys with various views and lightings. We split the data into training and test sets with disjoint ids (each ~ 24 k, 25 ids). We formed each group by randomly selecting images of the same id but varying the view and lighting.

B Architecture Details

Each model used convolutional neural nets as the encoders (g , r , h , and s) and a deconvolutional neural net as the decoder (f). Each encoder had three convolution layers with 64 filters (kernel 5×5 ; stride 2; padding 2) followed by two fully connected layers (64 intermediate and 2 or 3 output units for g and r ; 100 intermediate and 100 units for h and s). The decoder f had two fully connected layers (102 or 103 output and 128 intermediate units) followed by three transposed convolutional layers with 64 filters (kernel 6×6 ; upsampling 2; cropping 2). An RELU layer was inserted between convolutional or fully connected layers; the encoders r and s had an additional nonlinearity layer after the top layer to ensure positivity: $F(a) = \exp(a/2)$. Only for an MLVAE model, we let the deep net s encode the precision instead of the variance.

References

- [Bouchacourt *et al.*, 2018] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-Level Variational Autoencoder: Learning Disentangled Representations from Grouped Observations. In *AAAI Conference on Artificial Intelligence*, 2018.
- [Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN - Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. *Advances in neural information processing systems*, 2016.
- [Chen *et al.*, 2018] Mickaël Chen, Ludovic Denoyer, and Thierry Artières. Multi-View Data Generation Without View Supervision. In *International Conference on Learning Representations*, November 2018.
- [Cheung *et al.*, 2014] Brian Cheung, Jesse A Livezey, Arjun K Bansal, and Bruno A Olshausen. Discovering Hidden Factors of Variation in Deep Networks. In *International Conference on Learning Representations, Workshop*, December 2014.
- [Denton and Birodkar, 2017] Emily Denton and Vighnesh Birodkar. Unsupervised Learning of Disentangled Representations from Video. *Advances in neural information processing systems*, May 2017.
- [Dosovitskiy and Springenberg, 2015] A Dosovitskiy and J Tobias Springenberg. Learning to generate chairs with convolutional neural networks. *Computer Vision and Pattern Recognition*, 2015.
- [El Shafey *et al.*, 2013] Laurent El Shafey, Chris McCool, Roy Wallace, and Sébastien Marcel. A scalable formulation of probabilistic linear discriminant analysis: applied to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1788–1794, July 2013.
- [Földiák, 1991] P Földiák. Learning invariance from transformation sequences. *Neural Computation*, 3(2):194–200, 1991.
- [Goodfellow *et al.*, 2014] I Goodfellow, J Pouget-Abadie, and M Mirza. Generative adversarial nets. *Advances in neural information processing systems*, 2014.
- [Gross *et al.*, 2010] Ralph Gross, Iain Matthews, Jeff Cohn, Takeo Kanade, and Simon Baker. Multi-PIE. *Proceedings of the International Conference on Automatic Face and Gesture Recognition. IEEE International Conference on Automatic Face & Gesture Recognition*, 28(5):807–813, May 2010.
- [Higgins *et al.*, 2016] 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. In *International Conference on Learning Representations*, November 2016.
- [Hosoya and Hyvärinen, 2016] H Hosoya and A Hyvärinen. Learning Visual Spatial Pooling by Strong PCA Dimension Reduction. *Neural Computation*, 28:1249–1263, 2016.
- [Hosoya and Hyvärinen, 2017] Haruo Hosoya and Aapo Hyvärinen. A mixture of sparse coding models explaining properties of face neurons related to holistic and parts-based processing. *PLoS Computational Biology*, 13(7):e1005667, July 2017.
- [Kingma and Ba, 2015] D Kingma and J Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.
- [Kingma and Welling, 2014] D P Kingma and M Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.
- [Kingma *et al.*, 2014] D P Kingma, S Mohamed, and D J Rezende. Semi-supervised learning with deep generative models. *Advances in neural information processing systems*, 2014.
- [Lample *et al.*, 2017] Guillaume Lample, Neil Zeghidour, Nicolas Usunier, Antoine Bordes, Ludovic Denoyer, and Marc’Aurelio Ranzato. Fader Networks: Manipulating Images by Sliding Attributes. *Advances in neural information processing systems*, pages 5967–5976, 2017.
- [LeCun *et al.*, 2004] Yann LeCun, Fu Jie Huang, and Léon Bottou. Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. *Computer Vision and Pattern Recognition*, 2004.
- [Mathieu *et al.*, 2016] Michaël Mathieu, Junbo Jake Zhao, Pablo Sprechmann, Aditya Ramesh, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. *Advances in neural information processing systems*, 2016.
- [Reed *et al.*, 2015] Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep Visual Analogy-Making. *Advances in neural information processing systems*, 2015.
- [Schüldt *et al.*, 2004] Christian Schüldt, Ivan Laptev, and Barbara Caputo. Recognizing Human Actions - A Local SVM Approach. In *International Conference on Pattern Recognition*, pages 32–36 Vol.3. IEEE, 2004.
- [Siddharth *et al.*, 2017] N Siddharth, Brooks Paige, Jan-Willem van de Meent, Alban Desmaison, Frank Wood, Noah D Goodman, Pushmeet Kohli, and Philip H S Torr. Learning Disentangled Representations with Semi-Supervised Deep Generative Models. *Advances in neural information processing systems*, 2017.
- [Tenenbaum and Freeman, 2000] J B Tenenbaum and W T Freeman. Separating style and content with bilinear models. *Neural Computation*, 12(6):1247–1283, June 2000.
- [Wang *et al.*, 2017] Chaoyue Wang, Chaohui Wang, Chang Xu, and Dacheng Tao. Tag Disentangled Generative Adversarial Networks for Object Image Re-rendering. *International Joint Conference on Artificial Intelligence*, August 2017.
- [Yang *et al.*, 2015] Jimei Yang, Scott Reed, Ming-Hsuan Yang, and Honglak Lee. Weakly-supervised Disentangling with Recurrent Transformations for 3D View Synthesis. *Advances in neural information processing systems*, 2015.