Deformable Generator Network: Unsupervised Disentanglement of Appearance and Geometry

Xianglei Xing[⋆] [1] Ruiqi Gao [2] Tian Han [2] Song-Chun Zhu [2] Ying Nian Wu [2]

Abstract

We propose a deformable generator model to disentangle the appearance and geometric information from images into two independent latent vectors. The appearance generator produces the appearance information, including color, illumination, identity or category, of an image. The geometric generator produces displacement of the coordinates of each pixel and performs geometric warping, such as stretching and rotation, on the appearance generator to obtain the final synthesized image. The proposed model can learn both representations from image data in an unsupervised manner. The learned geometric generator can be conveniently transferred to the other image datasets to facilitate downstream AI tasks.

1. Introduction

A fundamental challenge in developing a conceptual understanding of our world is learning the factorial structure of the observations without supervision (Bengio et al., 2013; Mathieu et al., 2016). Conceptual understanding requires a disentangled representation which separates the underlying explanatory factors and explicitly represents the important attributes of the real-world data (Achille & Soatto, 2017; Burgess et al., 2018). For instance, given an image dataset of human faces, a disentangled representation can include the face's appearance attributes, such as color, light source, identity, gender, and the geometric attributes, such as face shape and viewing angle. A disentangled representation is useful not only for building more transparent and interpretable generative models, but also for a large variety of downstream AI tasks such as transfer learning and zero-shot inference where humans excel but machines struggle (Lake et al., 2017). Many exciting applications require generative models that can synthesize novel instances while certain key factors of variation are held fixed. Potential applications include generating a face image with desired attributes, such as color, face shape, expression and view, or transferring the face shape, expression, or view learned from one person to another person.

Generative models have shown great promise in learning disentangled representations of images. The generative models used for unsupervised disentangling usually fall into two categories: the Generative Adversarial Net (GAN) framework (Denton et al., 2015; Goodfellow et al., 2014; Li et al., 2018; Radford et al., 2016; Tran et al., 2017) and the Variational Autoencoder (VAE) framework (Kingma & Welling, 2013; Kumar et al., 2017; Mnih & Gregor, 2014; Rezende et al., 2014). InfoGAN (Chen et al., 2016), a representative of the former family, is motivated by the principle of the maximization of the mutual information between the observations and a subset of latent vectors. However, its disentangling performance is sensitive to the choice of the prior and the number of latent vectors. The β-VAE (Higgins et al., 2016), from the latter family, learns disentangled representations by utilizing a VAE objective with an extra KL penalty to encourage the latent distribution (variational posterior) to be close to the standard normal distribution, giving a more robust and stable solution for disentangling.

In contrast to the existing methods which use one latent vector to encode the factors of variation, our work introduces a deformable generator network that disentangles the appearance and geometric information from an image into two independent latent vectors in an unsupervised manner. Motivated by the Active Appearance Models (AAM) (Cootes et al., 2001; Kossaifi et al., 2017b) which uses a linear model for jointly capturing the appearance and shape variation in an image, the proposed model introduces two nonlinear generators to extract the appearance and geometry information separately. Unlike the AAM method (Cootes et al., 2001; Kossaifi et al., 2017a;b) which requires hand-annotated fa-

[1] College of Automation, Harbin Engineering University, Heilongjiang, China [2] Department of statistics, University of California, Los Angeles, California, USA. Correspondence to: Xianglei Xing <xingxl@hrbeu.edu.cn>, Yingnian Wu <ywu@stat.ucla.edu>.

Presented at the Proceedings of the 35^{th} International Conference on Machine Learning workshop on Theoretical Foundations and Applications of Deep Generative Models, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

[⋆] The work was done while the author worked as a visiting scholar at UCLA.

cial landmarks for each training image, the proposed deformable generator model is purely unsupervised and learns from images alone.

2. Model and learning algorithm

This section provides the details of the model and the associated learning and inference algorithm.

2.1. Model

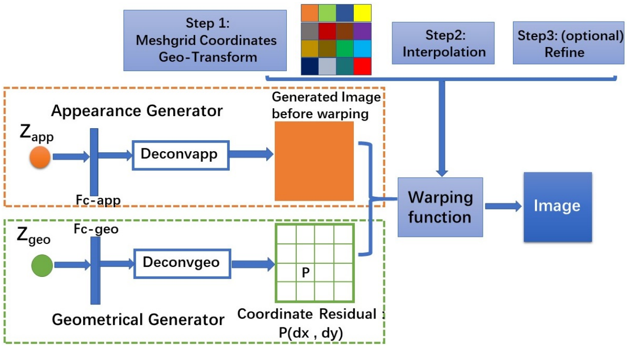


Figure 1. An illustration of the proposed model. The model contains two generator networks: one appearance generator and one geometric generator. The two generators are connected with a warping function to produce the final image. The warping function includes a geometric transformation operation for image coordinates and a differentiable interpolation operation. The refining operation is optional for improving the warping function.

The proposed model contains two generator networks: one appearance generator and one geometric generator. The two generators are connected with a warping function to produce the final images or video frames, as shown in Figure 1. Suppose an arbitrary image or video frame $X \in \mathbb{R}^{D_x \times D_y \times 3}$ is generated with two independent latent vectors, $Z^a \in \mathbb{R}^{d_a}$ which controls its appearance, and $Z^g \in \mathbb{R}^{d_g}$ which controls its geometric information. Varying the geometric latent vector Z^g and fixing the latent vector Z^a of appearance, we can transform an object's geometric information, such as rotating it with some angles and changing its shape. Varying the Z^a and fixing the Z^g, we can change the identity or the category of the object, while keeping it within the same geometric status, such as the same viewing angle or the same shape. Thus, the appearance information and the geometric information are disentangled in the ideal situation.

The model can be expressed as

$$\begin{aligned} X &= F(Z^a, Z^g; \theta) \\ &= F_w(F_a(Z^a; \theta_a), F_g(Z^g; \theta_g)) + \epsilon \qquad (1) \end{aligned}$$

where $Z^a \sim \mathrm{N}(0, I_{d_a})$, $Z^g \sim \mathrm{N}(0, I_{d_g})$, and $\epsilon \sim \mathrm{N}(0, \sigma^2 I_D)$ ($D = D_x \times D_y \times 3$) are independent. F_w is the warping function, which employs the features generated by the geometric generator $F_g(Z^g; \theta_g)$ to warp the geometry of the image from the appearance generator $F_a(Z^a; \theta_a)$ to obtain the final output image X.

2.2. Warping function

A warping function usually includes a geometric transformation operation for image coordinates and a differentiable interpolation (or resampling) operation. The geometric transformation describes the destination coordinates (x, y) for every location (u, v) in the source coordinates. The geometric operation only modifies the positions of pixels in an image without changing their colors and illumination. Therefore, the color and illumination information and the geometric information are naturally disentangled by the geometric generator and the appearance generator in the proposed model.

The geometric transformation Φ can be a rigid affine mapping, as is used in the spatial transformer networks (Jaderberg et al., 2015), or a non-rigid deformable mapping, which is the case in our work. Specifically, the coordinates displacement (dx, dy) (or the dense optical flow field) of each regular grid (x, y) in the output warping image X are generated by our geometric generator $F_g(Z^g; \theta_g)$. The point-wise transformation in this deformable mapping can be formulated as

$$\begin{pmatrix} u \\ v \end{pmatrix} = \Phi_{(Z^g, \theta_g)} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x + dx \\ y + dy \end{pmatrix} \qquad (2)$$

where (u, v) are the source coordinates of the image generated by the appearance generator $F_a(Z^a; \theta_a)$.

Since the evaluated (u, v) by Eq.(2) does not usually have integer coordinates, each pixel's value of the output warping image X can be computed by a differentiable interpolation operation. Let $X_a = F_a(Z^a; \theta_a)$ denotes the image generated by the appearance generator. The warping function F_w can be formulated as

$$X(x, y) = F_i(X_a(x + dx, y + dy)), \qquad (3)$$

where F_i is the differentiable interpolation function. In this study, we employ a differentiable bilinear interpolation of the form

$$X(x, y) = \sum_{j}^{D_y} \sum_{i}^{D_x} X_a(i, j) M(1 - |u - i|) M(1 - |v - j|) \qquad (4)$$

where $M(\cdot) = \max(0, \cdot)$, and from Eq.(2), we have $u = x + dx$,$v = y + dy$. The details of back-propagation through this bilinear interpolation can be found in (Jaderberg et al., 2015).

The displacement (dx, dy) is used in the deformable convolutional networks (Dai et al., 2017). The computation of coordinates displacement (dx, dy) is known as the optical flow estimation (Brox et al., 2004; Dosovitskiy et al., 2015; Horn & Schunck, 1981; Ilg et al., 2017; Ranjan & Black, 2017; Sun et al., 2014). Our work is concerned with modeling and generating the optical flow, in addition to estimating the optical flow.

The displacement (dx, dy) can be caused by the motion of the objects in the scene. It can also be caused by the change of viewpoint relative to 3D objects in the scene. Therefore, it is natural to incorporate motion and 3D models into the geometric generator where the change or variation of Z^g depends on the motion and 3D information.

2.3. Inference and learning

To learn this deformable generator model, we introduce a learning and inference algorithm for two latent vectors, without designing and learning extra inference networks. Our method is motivated by a maximum likelihood learning algorithm for generator networks (Han et al., 2017). Specifically, the proposed model can be trained by maximizing the log-likelihood on the training dataset $\{X_i, i = 1, \dots, N\}$,

$$\begin{aligned} L(\theta) &= \frac{1}{N}\sum_{i=1}^{N} \log p(X_i; \theta) \\ &= \frac{1}{N}\sum_{i=1}^{N} \log \int p(X_i, Z_i^a, Z_i^g; \theta) dZ_i^a dZ_i^g. \quad (5) \end{aligned}$$

The uncertainties in inferring Z_i^a and Z_i^g are taken into account by the above observed-data log-likelihood.

We can evaluate the gradient of $L(\theta)$ according to the following well-known result which is related to the EM algorithm:

$$\begin{aligned} &\frac{\partial}{\partial\theta} \log p(X; \theta) \\ &= \frac{1}{p(X;\theta)} \frac{\partial}{\partial\theta} \int p(X, Z^a, Z^g) dZ^a dZ^g \\ &= \mathrm{E}_{p(Z^a, Z^g | X; \theta)} \left[\frac{\partial}{\partial\theta} \log p(X, Z^a, Z^g; \theta) \right] \quad (6) \end{aligned}$$

Since the expectation in Eq.(6) is usually analytically intractable, we employ Langevin dynamics to draw samples from the posterior $p(Z_a, Z_g | X; \theta)$ and compute the Monte Carlo average to obtain an approximation. For each observation X, the latent vectors Z^a and Z^g can be sampled from $p(Z^a, Z^g | X; \theta)$ alternately by Langevin dynamics: fixing Z^g and sampling Z^a from $p(Z^a | X; Z^g, \theta) \propto p(X, Z^a; Z^g, \theta)$, then fixing Z^a and sampling Z^g from $p(Z^g | X; Z^a, \theta) \propto p(X, Z^g; Z^a, \theta)$. The latent vectors are inferred and updated as follows:

$$\begin{aligned} Z_{t+1}^a &= Z_t^a + \frac{\delta^2}{2} \frac{\partial}{\partial Z^a} \log p(X, Z_t^a; Z_t^g, \theta) + \delta \mathcal{E}_t^a \\ Z_{t+1}^g &= Z_t^g + \frac{\delta^2}{2} \frac{\partial}{\partial Z^g} \log p(X, Z_t^g; Z_t^a, \theta) + \delta \mathcal{E}_t^g \quad (7) \end{aligned}$$

where t is the number of steps of the Langevin sampling, $\mathcal{E}$ is standard Gaussian noise added to prevent the chain from becoming trapped in local modes, and δ is the step size of Langevin dynamics. The log of the joint density in Eq.(7) can be evaluated by

$$\begin{aligned} &\log p(X, Z^a; Z^g, \theta) = \log\left[p(Z^a) p(X | Z^a, Z^g, \theta)\right] \\ &= -\frac{1}{2\sigma^2} \|X - F(Z^a, Z^g; \theta)\|^2 - \frac{1}{2}\|Z^a\|^2 + C_1 \\ &\log p(X, Z^g; Z^a, \theta) = \log\left[p(Z^g) p(X | Z^a, Z^g, \theta)\right] \\ &= -\frac{1}{2\sigma^2} \|X - F(Z^a, Z^g; \theta)\|^2 - \frac{1}{2}\|Z^g\|^2 + C_2 \quad (8) \end{aligned}$$

where F and σ are defined in Eq.(1), and both C_1 and C_2 are constants. It can be shown that, given sufficient transition steps, the Z^a and Z^g obtained from this procedure follow their joint posterior distribution.

Obtaining independent samples of the posterior density in each training iteration is infeasible due to the high computational cost of the MCMC updates. In this paper, the MCMC transitions of both Z^a and Z^g start from the updated latent vectors from the previous learning iteration. The persistent updating results in a chain that is long enough to sample from the posterior distribution, and the warm initialization vastly reduces the computational burden of the MCMC updates. The convergence of stochastic gradient descent based on persistent MCMC has been studied by (Younes, 1999).

For each training example X_i, we run the Langevin dynamics in Eq.(7) to get the corresponding posterior sample Z_i^a and Z_i^g. The sample is then used for gradient computation in Eq.(6). More precisely, the parameter θ is learned through Monte Carlo approximation:

$$\begin{aligned} \frac{\partial}{\partial\theta} L(\theta) &\approx \frac{1}{N}\sum_{i=1}^{N} \frac{\partial}{\partial\theta} \log p(X_i, Z_i^a, Z_i^g; \theta) \\ = \frac{1}{N}\sum_{i=1}^{N} &\frac{1}{\sigma^2}(X_i - F(Z_i^a, Z_i^g; \theta)) \frac{\partial}{\partial\theta} F(Z_i^a, Z_i^g; \theta). \quad (9) \end{aligned}$$

The whole algorithm iterates through two steps: (1) inferential step which infers the latent vectors through Langevin dynamics, and (2) learning step which learns the network parameters θ by stochastic gradient descent. Gradient computations in both steps are powered by back-propagation. Algorithm 1 describes the details of the learning and inference algorithm.

The model can also be learned by VAE (Kingma & Welling, 2013) by designing an extra inference network to infer (Z^a, Z^g). We have implemented VAE, and our preliminary results suggest that it can learn meaningful deformable generator. The model can also be learned by GAN (Goodfellow et al., 2014) by designing an extra discriminator network.

```
Algorithm 1 Learning and inference algorithm
Require:
  (1) training examples $\{X_i \in \mathbb{R}^{D_x \times D_y \times 3}, i = 1, \dots, N\}$
  (2) number of Langevin steps $l$
  (3) number of learning iterations $T$
Ensure:
  (1) learned parameters $\theta$
  (2) inferred latent vectors $\{Z_i^a, Z_i^g, i = 1, \dots, N\}$

  1: Let $t \leftarrow 0$, initialize $\theta$.
  2: Initialize $\{Z_i^a, Z_i^g, i = 1, \dots, N\}$
  repeat
    3: Inference back-propagation: For each $i$, run $l$ steps of Langevin dynamics to alternatively sample $Z_i^a$ from $p(Z_i^a|X_i; Z_i^g, \theta)$, while fixing $Z_i^g$; and sample $Z_i^g$ from $p(Z_i^g|X_i; Z_i^a, \theta)$, while fixing $Z_i^a$. Starting from the current $Z_i^a$ and $Z_i^g$, each step follows Eq.(7).
    4: Learning back-propagation: Update $\theta_{t+1} \leftarrow \theta_t + \eta_t L'(\theta_t)$, with learning rate $\eta_t$, where $L'(\theta_t)$ is computed according to Eq.(9).
    5: Let $t \leftarrow t + 1$
  until $t = T$
```

In this work, we use the current learning and inference algorithm mainly for the sake of simplicity, so that we do not need to design and learn extra networks.

3. Experiments

In this section, we first qualitatively demonstrate that our proposed deformable generator framework consistently disentangles the appearance and geometric information. We then analyze and evaluate the proposed model quantitatively. The deformable generator network structures and parameters are listed in the Appendix. We set the value of the interpolation parameter γ to 10 in the experiments, i.e., we vary the components of the latent vectors within the range $[-\gamma, \gamma]$ when visualizing the effects of the components.

3.1. Qualitative experiments

3.1.1. Experiments on CelebA

We first train the deformable generator on the 10,000 images from CelebA benchmark dataset (Liu et al., 2015). Some examples in CelebA are shown in Figure 2, which are processed by the OpenFace (Amos et al., 2016) and cropped to 64 × 64 pixels. To study the performance of the proposed method for disentangling the appearance and geometric information, we investigate the effect of different combinations of the geometric latent vector Z^g and the appearance latent vector Z^a. (1) Set the geometric latent vector Z^g to zero, and each time vary one dimension of the appearance variable Z^a from $[-\gamma, \gamma]$ with a uniform step $\frac{2\gamma}{10}$, while holding the other dimensions of Z^a at zero. Some typical generated images are shown in Figure 3. (2) Set Z^a to be a fixed value, and each time vary one dimension of the geometric latent vector Z^g from $[-\gamma, \gamma]$ with a uniform step $\frac{2\gamma}{10}$, while keeping the other dimensions of Z^g at zero. Some representative generated results are shown in Figure 4.

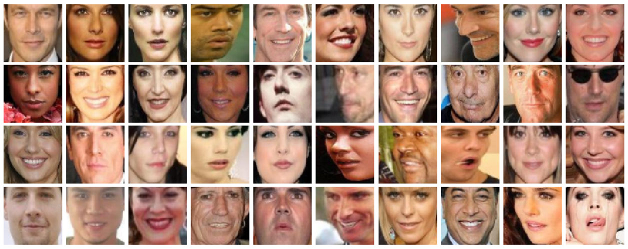


Figure 2. Example training images from CelebA. The training set contains 10000 images from CelebA, and they are cropped to 64×64 pixels by the OpenFace. These faces have different colors, illuminations, identities, viewing angles, shapes, and expressions.

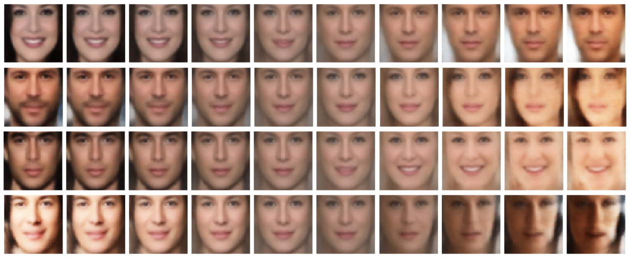


Figure 3. Each dimension of the appearance latent vector encodes appearance information such as color, illumination and gender. In the first line, from left to right, the color of background varies from black to white, and the gender changes from a woman to a man. In the second line, the moustache of the man becomes thicker when the corresponding dimension of Z^a approahces zero, and the hair of the woman becomes denser when the corresponding dimension of Z^a increases. In the third line, from left to right, the skin color changes from dark to white. In the fourth line, from left to right, the illumination lighting changes from the left-side of the face to the right-side of the face.

As we can observe from Figure 3, (1) although the training faces from CelebA have different viewing angles, the appearance latent vector only encodes front-view information, and (2) each dimension of the appearance latent vector encodes appearance information such as color, illumination and identity. For example, in the first line of Figure 3, from left to right, the color of background varies from black to white, and the identity of the face changes from a women to a man. In the second line of Figure 3, the moustache of

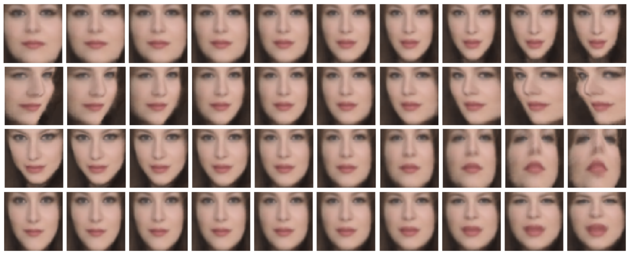
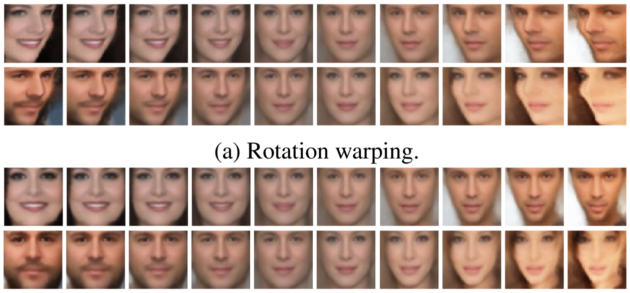


Figure 4. Each dimension of the geometric latent vector encodes fundamental geometric information such as shape and viewing angle. In the first line, the shape of the face changes from fat to thin from left to the right. In the second line, the pose of the face varies from left to right. In the third line, from left to right, the vertical tilt of the face varies from downward to upward. In the fourth line, the face width changes from stretched to cramped.



(a) Rotation warping.

(b) Shape warping.

Figure 5. Applying the (a) rotation warping and (b) shape warping operations learned by the geometric generator to the canonical faces generated by the appearance generator. Compared with Figure 3, only the pose information varies, and the identity information is kept in the process of warping.

the man becomes thicker when the value of the corresponding dimension of Z^a decreases, and the hair of the woman becomes denser when the value of the corresponding dimension of Z^a increases. In the third line, from left to right, the skin color varies from dark to white, and in the fourth line, from left to right, the illumination lighting changes from the left-side of the face to the right-side of the face.

From Figure 4, we have the following interesting observations. (1) The geometric latent vectors does not encode any appearance information. The color, illumination and identity are the same across these generated images. (2) Each dimension of the geometric latent vector encodes fundamental geometric information such as shape and viewing angle. For example, in the first line of Figure 4, the shape of the face changes from fat to thin from left to the right; in the second line, the pose of the face varies from left to right; in the third line, from left to right, the tilt of the face varies from downward to upward; and in the fourth line, the expression changes from stretched to cramped.

From the results in Figures 3 and 4, we find that the appearance and geometric information of face images have been disentangled effectively. Therefore, we can apply the geometric warping (e.g. operations in Figure 4) learned by the geometric generator to all the canonical faces (e.g. generated faces in Figure 3) learned by the appearance generator. Figure 5 demonstrates the effect of applying geometric warping to the generated canonical faces in Figure 3. Comparing Figure 3 with Figure 5, we find that the rotation and shape warping operations do not modify the identity information of the canonical faces, which corroborates the disentangling power of the proposed deformable generator model.

Furthermore, we evaluate the disentangling ability of the proposed model by transferring and recombining geometric and appearance vectors from different faces. Specifically, we first feed 7 unseen images from CelebA into our deformable generator model to infer their appearance vectors Z_1^a, Z_2^a,. . . ,Z_7^a and geometric vectors Z_1^g, Z_2^g ,. . . ,Z_7^g using the Langevin dynamics (with 300 steps) in Eq.(7). Then, we transfer and recombine the appearance and geometric vectors and use $\{Z_1^a, Z_2^g\}$, . . . , $\{Z_1^a, Z_7^g\}$ to generate six new face images, as shown in the second row of Figure 6. We also transfer and recombine the appearance and geometric vectors and use $\{Z_2^a, Z_1^g\}$,. . . , $\{Z_7^a, Z_1^g\}$ to generate another six new faces, as shown in the third row of Figure 6. From the 2nd to the 7th column, the images in the second row have the same appearance vector Z^a, but the geometric latent vectors Z^g are swapped between each image pair. As we can observe from the second row of Figure 6, (1) the geometric information of the original images are swapped in the synthesized images, and (2) the inferred Z^g can capture the view information of the unseen images. The images in the third row of Figure 6 have the same geometric vector Z_1^g, but the appearance vectors Z^a are swapped between each image pair. From the third row of Figure 6, we observe that (1) the appearance information are exchanged. (2) The inferred Z^a capture the color, illumination and coarse appearance information but lose more nuanced identity information. Only finite features are learned from 10k CelebA images, and the model may not contain the features necessary to closely model an unseen face.

The learned geometric information can be directly applied to the faces of animals such as cats and monkeys, as shown in Figure 7. The monkey and cat faces rotate from left to right when the rotation warping learned from human faces is applied. The shape of both the monkey and cat faces changes from fat to thin when the shape warping learned by the geometric generators is used.

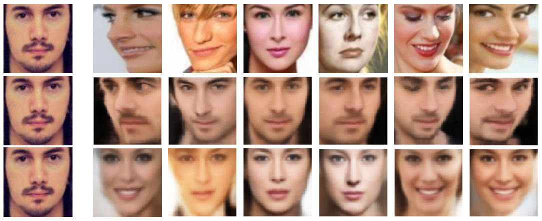


Figure 6. Transferring and recombining geometric and appearance vectors. The first row shows the 7 unseen faces from CelebA. The second row shows the generated faces by transferring and recombining the first row's 2th-7th faces' geometric vectors with the first row's 1th face's appearance vector. The third row shows the generated faces by transferring and recombining the first row's 2th-7th faces' appearance vectors with the first row's 1th face's geometric vector.

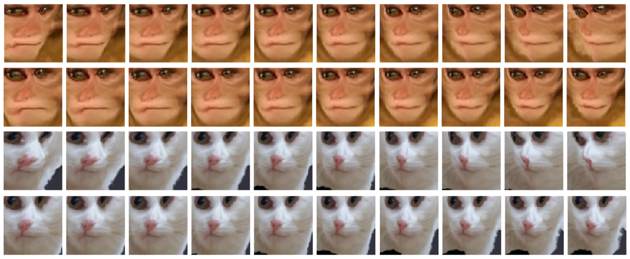
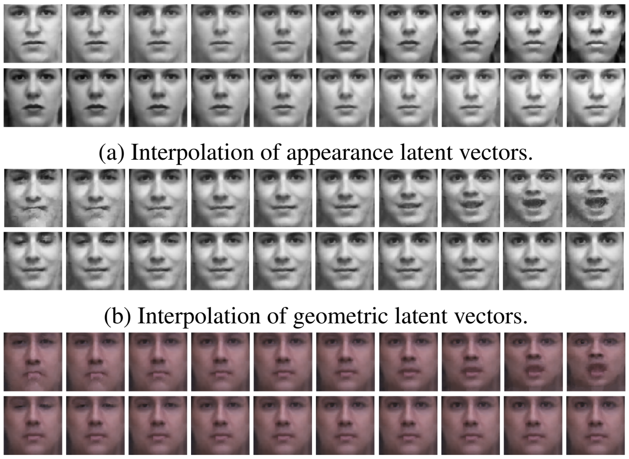


Figure 7. Applying the learned geometric warping from CelebA to animal faces. The first two rows show a monkey face after applying the rotation and shape warping learned from CelebA. The third and the fourth rows show a cat face after applying the rotation and shape warping learned from CelebA.

3.1.2. Experiments on expression dataset

We next study the performance of the proposed deformable generator model on the face expression dataset CK+ (Lucey et al., 2010). Following the same experimental protocol as the last subsection, we can investigate the change produced by each dimension of the appearance latent vector (after setting the value of geometric latent vector to zero) and the geometric latent vector (after setting the appearance latent vector to a fixed value). The disentangled results are shown in Figure 8. The training faces from CK+ have labels of expressions, but we do not use any such labels in our unsupervised learning method. Although the dataset contains faces of different expressions, the learned appearance latent vector usually encodes a neutral expression. The geometric latent vector controls major variation in expression, but does not change the identity information.

To test whether appearance and geometric information are disentangled in the proposed model, we try to transfer the learned expression from CK+ to another face dataset, Multi-Pie (Gross et al., 2010), by fine-turning the appearance generator on the target face dataset while fixing the parameters in the geometric generator. Figure 8(c) shows the result of transferring the expressions of 8(b) into the faces of Multi-Pie. The expressions from the gray faces of CK+ have been transferred into the color faces of Multi-Pie.



(a) Interpolation of appearance latent vectors.

(b) Interpolation of geometric latent vectors.

(c) Transfering the expression in (b) to the face images in Multi-PIE dataset.

Figure 8. Interpolation examples of (a) appearance latent vectors and (b) geometric latent vectors. (c) Transfer the learned expression to the face images in Multi-PIE dataset.

3.1.3. Experiment on CIFAR-10

We further test our model on the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset, which includes various object categories and has 50,000 training examples. We randomly sample Z^a from $\mathrm{N}(0, \mathbf{I}_{d_a})$. For Z^g, we interpolate one dimension from $-\gamma$ to γ and fix the other dimensions to 0. Figure 9 shows interpolated examples generated by model learned from the *car* category. For each row, we use different Z^a and interpolate the same dimension of Z^g. The results show that each dimension of Z^g controls a specific geometric transformation.

3.2. Quantitative experiments

3.2.1. Covariance between the latent vectors and geometric variation

To quantitatively study the covariance between each dimension of the latent vectors (Z^g, Z^a) and input images with geometric variation, we select images with ground-truth labels that record geometric attributes, specifically the multi-view face images from the Multi-Pie dataset (Gross et al., 2010). For each of the 5 viewing angles $\{-30°, -15°, 0°, 15°, 30°\}$, we feed 100 images into the learned model to infer their geometric latent vector Z^g and appearance latent vector Z^a. Under each view

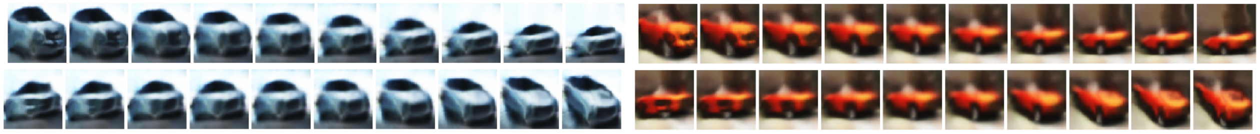


Figure 9. Synthesized examples generated by the model learned from the *car* category of CIFAR-10 dataset. For each row, we interpolate the same dimension of Z^g from $-\gamma$ to $+\gamma$, and fix the other dimensions of Z^g to zero.

$\theta \in \{-30°, -15°, 0°, 15°, 30°\}$, we compute the means $\bar{Z}^g_\theta$ and $\bar{Z}^a_\theta$ of the inferred latent vectors. For each dimension i of Z^g, we construct a 5-dimensional vector $\bar{Z}^g(i) = [\bar{Z}^g_{-30°}(i), \bar{Z}^g_{-15°}(i), \bar{Z}^g_{0°}(i), \bar{Z}^g_{15°}(i), \bar{Z}^g_{30°}(i)]$. Similarly, we construct a 5-dimensional vector $\bar{Z}^a(i)$ under each dimension of Z^a. We normalize the viewing angles vector $\theta = [-30, -15, 0, 15, 30]$ to have unit norm. Finally, we compute the covariance between each dimension of the latent vectors (Z^g, Z^a) and input images with view variations as follows:

$$R^g_i = |\bar{Z}^g(i)^\top \theta|, \quad R^a_i = |\bar{Z}^a(i)^\top \theta| \tag{10}$$

where i denotes the i-th dimension of latent vector Z^g or Z^a, and $|\cdot|$ denotes the absolute value. We summarize the the covariance responses R^g and R^a of the geometric and appearance latent vectors in Figure 10. As we can observe in Figure 10, the R^g tends to be much larger than R^a.

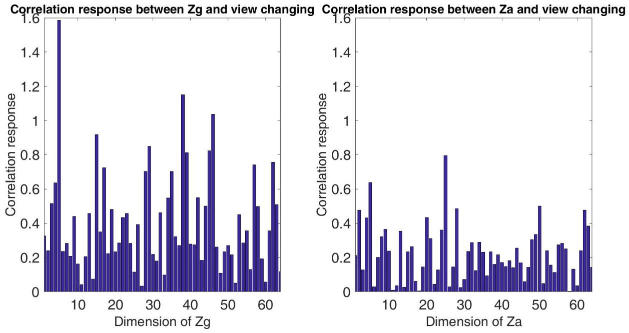
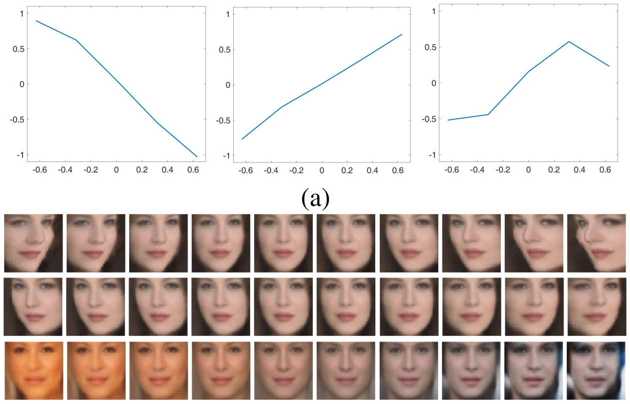


Figure 10. Absolute value of covariance between each dimension of the geometric (or appearance) latent vectors and view variations for the face images from Multi-Pie. The left subfigure shows covariance with the geometric latent vector; the right subfigure shows covariance with the appearance latent vector.

Moreover, for the two largest R^g_i and the largest R^a_i, we plot covariance relationship between the latent vector $\bar{Z}^g(i)$ (or $\bar{Z}^a(i)$) and viewing angles vector θ in Figure 11. As we can observe from the left and the center sub-figures from Figure 11, the $\bar{Z}^g(i)$ corresponding to the two largest R^g_i (R^g_5, R^g_{38}) have very strong negative and positive covariance respectively with change in viewing angle. However, as shown in the right sub-figure, the $\bar{Z}^a(i)$ corresponding to the largest R^a_i (R^a_{25}) does not have strong covariance with the change of viewing angle. We wish to point out that we should not expect Z^a to encode the identity exclusively and Z^g to encode the view exclusively, because different persons may have shape changes, and different views may have lighting or color changes.



(a)

(b)

Figure 11. (a) Covariance relationship between the mean latent vector $\bar{Z}^g(i)$ (or $\bar{Z}^a(i)$) and viewing angles vector θ. We choose two dimensions of Z^g (Z^g_5 and Z^g_{38}, left and center) with the largest covariance and one dimension of Z^a with the largest covariance (Z^a_{25}, right). (b) Images generated by varying the values of the three dimensions in (a) respectively, while fixing the values of other dimensions to be zero.

Furthermore, we generate face images by varying the dimension of Z^g corresponding to the two largest covariance responses from values $[-\gamma, +\gamma]$ with a uniform step $\frac{2\gamma}{10}$, while holding the other dimensions of Z^g to zero as we did in the subsection 4.1.1. Similarly, we generate face images by varying the dimension of Z^a corresponding to the largest covariance responses from values $[-\gamma, +\gamma]$ with a uniform step $\frac{2\gamma}{10}$, while holding the other dimensions of Z^a to zero. The generated images are shown in Figure 11(b). We can make several important observations. (1) The variation in viewing angle in the first two rows is very obvious, and the magnitude of the change in view in the first row is larger than that in the second row. This is consistent with the fact that $R^g_5 > R^g_{38}$ and with the observation that the slope in the left subfigure of Figure 11(a) is steeper than that of the center subfigure of Figure 11(a). (2) In the first row, the faces rotate from right to left and the covariance relationship in

Methods	VAE	ABP	Ours
Reconstruction Error	89.02	94.66	76.52

Table 1. The Mean Square Reconstruction Errors per image for unseen multi-view faces from the Multi-Pie dataset.

the left subfigure of Figure 11(a) is nearly perfect negative covariance. In the second row, the faces rotate from left to right and the covariance relationship in the center subfigure of Figure 11(a) is nearly perfect positive covariance. (3) It is difficult to find obvious variation in viewing angle in the third row. Therefore, these generated images further verify that the geometric generator of the proposed model mainly captures geometric variation, while the appearance generator is not sensitive to geometric variation.

3.2.2. Reconstruction error on unseen multi-view faces

Since the proposed deformable generator model can disentangle the appearance and geometric information from an image, we can transfer the geometric warping operation learned from one dataset into another dataset. Specifically, given 1000 front-view faces from the Multi-Pie dataset (Gross et al., 2010), we can fine-tune the appearance generator's parameters while fixing the geometric generator's parameters, which are learned from the CelebA dataset. Then we can reconstruct unseen images that have various viewpoints. In order to quantitatively evaluate the geometric knowledge transfer ability of our model, we compute the reconstruction error on 5000 unseen images from Multi-Pie for the views $\{-30°, -15°, 0°, 15°, 30°\}$, with 1000 faces for each view. We compare the proposed model with the state-of-art generative models, such as VAE (Burgess et al., 2018; Kingma & Welling, 2013) and ABP (Han et al., 2017). For fair comparison, we first train the VAE and ABP models with the same CelebA training set of 10,000 faces, and then fine-tune them on the 1000 front-view faces from the Multi-Pie dataset. The mean square reconstruction error per image for each method is shown in Table 1. As we can observe from Table 1, the proposed method obtains the lowest reconstruction error. Our model benefits from the transferred geometric knowledge learned from the CelebA dataset, while both the VAE and ABP models cannot efficiently learn or transfer purely geometric information.

3.3. Balancing explaining-away competition

The proposed deformable generator model utilizes two generator networks to disentangle the appearance and geometric information from an image. Since the geometric generator only produces displacement for each pixel without modifying the pixel's value, the color and illumination information and the geometric information are naturally disentangled by the proposed model's specific structure.

In order to properly disentangle the identity (or category) and the view (or geometry) information, the learning capacity between the appearance generator and geometric generator should be balanced. The appearance generator and the geometric generator cooperate with each other to generate the images. Meanwhile, they also compete against each other to explain away the training images. If the learning of the appearance generator outpaces that of the geometric generator, the appearance generator will encode most of the knowledge (including the view and shape information), while the geometric generator will only learn minor warping operations. On the other hand, if the geometric generator learns much faster than the appearance generator, the geometric generator will encode most of the knowledge (including the identity or category information, which should be encoded by the appearance network).

To control the tradeoff between the two generators, we introduce a balance parameter α, which is defined as the ratio of the number of filters within each layer between the appearance and geometric generators. The balance parameter α should not be too large or too small. We set α to 0.625 in our experiments.

4. Conclusion

In this study, we propose a deformable generator model which aims to disentangle the appearance and geometric information of an image into two independent latent vectors Z_a and Z_g. The learned geometric generator can be transferred to other datasets, or can be used for the purpose of data augmentation to produce more variations in the training data for better generalization.

In addition to the learning and inference algorithm adopted in this paper, the model can also be trained by VAE and GAN, as well as their generalizations such as β-VAE and info-GAN, which target disentanglement in general.

The model can be generalized to a dynamic one by adding transition models for the latent vectors. While the transition model for the appearance vector may generates dynamic textures of non-trackable motion, the transition model for the geometric vector may generate intuitive physics of trackable motion. The geometric generator can also be generalized to incorporate 3D information of rigid or non-rigid 3D objects.

The warping function based on coordinate displacements is hand designed. A refinement of this warping function in the form of a residual in addition to the warping function may be learned. However, we tend to believe that the warping function itself or more importantly the notion of coordinate displacements may have to be a fundamentally innate part of a model for vision that may not be learned from the data.

Acknowledgment

This work was supported by DARPA SIMPLEX N66001-15-C-4035, ONR MURI N00014-16-1-2007, DARPA ARO W911NF-16-1-0579, DARPA N66001-17-2-4029, Natural Science Foundation of China No. 61703119, Natural Science Fund of Heilongjiang Province of China No. QC2017070 and Fundamental Research Funds for the Central Universities of China No. HEUCFM180405. We thank Mitchell K. Hill for his assistance with writing.

References

Achille, Alessandro and Soatto, Stefano. Emergence of invariance and disentanglement in deep representations. In *Proc. International Conference on Machine Learning (ICML), Sydney*, 2017.

Amos, Brandon, Ludwiczuk, Bartosz, and Satyanarayanan, Mahadev. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

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

Brox, Thomas, Bruhn, Andrés, Papenberg, Nils, and Weickert, Joachim. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pp. 25–36. Springer, 2004.

Burgess, Christopher P, Higgins, Irina, Pal, Arka, Matthey, Loic, Watters, Nick, Desjardins, Guillaume, and Lerchner, Alexander. Understanding disentangling in beta-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Chen, Xi, Duan, Yan, Houthooft, Rein, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in Neural Information Processing Systems*, pp. 2172–2180, 2016.

Cootes, Timothy F., Edwards, Gareth J., and Taylor, Christopher J. Active appearance models. *IEEE Transactions on pattern analysis and machine intelligence*, 23(6):681–685, 2001.

Dai, Jifeng, Qi, Haozhi, Xiong, Yuwen, Li, Yi, Zhang, Guodong, Hu, Han, and Wei, Yichen. Deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 764–773, 2017.

Denton, Emily L, Chintala, Soumith, Fergus, Rob, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pp. 1486–1494, 2015.

Dosovitskiy, Alexey, Fischer, Philipp, Ilg, Eddy, Hausser, Philip, Hazirbas, Caner, Golkov, Vladimir, van der Smagt, Patrick, Cremers, Daniel, and Brox, Thomas. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2758–2766, 2015.

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

Gross, Ralph, Matthews, Iain, Cohn, Jeffrey, Kanade, Takeo, and Baker, Simon. Multi-pie. *Image Vision Comput.*, 28 (5):807–813, May 2010. ISSN 0262-8856.

Han, Tian, Lu, Yang, Zhu, Song-Chun, and Wu, Ying Nian. Alternating back-propagation for generator network. In *AAAI*, pp. 1976–1984, 2017.

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

Horn, Berthold KP and Schunck, Brian G. Determining optical flow. *Artificial intelligence*, 17(1-3):185–203, 1981.

Ilg, Eddy, Mayer, Nikolaus, Saikia, Tonmoy, Keuper, Margret, Dosovitskiy, Alexey, and Brox, Thomas. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

Jaderberg, Max, Simonyan, Karen, Zisserman, Andrew, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pp. 2017–2025, 2015.

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

Kossaifi, Jean, Tran, Linh, Panagakis, Yannis, and Pantic, Maja. Gagan: Geometry-aware generative adverserial networks. *arXiv preprint arXiv:1712.00684*, 2017a.

Kossaifi, Jean, Tzimiropoulos, Georgios, and Pantic, Maja. Fast and exact newton and bidirectional fitting of active appearance models. *IEEE transactions on image processing*, 26(2):1040–1053, 2017b.

Krizhevsky, Alex and Hinton, Geoffrey. Learning multiple layers of features from tiny images. 2009.

Kumar, Abhishek, Sattigeri, Prasanna, and Balakrishnan, Avinash. Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848*, 2017.

Lake, Brenden M, Ullman, Tomer D, Tenenbaum, Joshua B, and Gershman, Samuel J. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

Li, Zejian, Tang, Yongchuan, and He, Yongxing. Unsupervised disentangled representation learning with analogical relations. *arXiv preprint arXiv:1804.09502*, 2018.

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Lucey, Patrick, Cohn, Jeffrey F, Kanade, Takeo, Saragih, Jason, Ambadar, Zara, and Matthews, Iain. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 94–101. IEEE, 2010.

Mathieu, Michael F, Zhao, Junbo Jake, Zhao, Junbo, Ramesh, Aditya, Sprechmann, Pablo, and LeCun, Yann. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pp. 5040–5048, 2016.

Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. In *ICML*, 2014.

Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

Ranjan, Anurag and Black, Michael J. Optical flow estimation using a spatial pyramid network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017.

Rezende, Danilo J., Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. In *NIPS*, pp. 1278–1286, 2014.

Sun, Deqing, Roth, Stefan, and Black, Michael J. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014.

Tran, Luan, Yin, Xi, and Liu, Xiaoming. Disentangled representation learning gan for pose-invariant face recognition. In *CVPR*, volume 3, pp. 7, 2017.

Younes, Laurent. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics: An International Journal of Probability and Stochastic Processes*, 65(3-4):177–228, 1999.

A. Deformable generator' Network Structure and Parameters

Table 2. Appearance generator's Network Structure.

Layers	In-Out Shape	Conv-Kernel size	stride
Z_a	64×1	Null	Null
Fc1	$4 \times 4 \times 80$	64×1280	Null
Deconv1	$8 \times 8 \times 40$	$3 \times 3 \times 80 \times 40$	2
Deconv2	$16 \times 16 \times 20$	$3 \times 3 \times 40 \times 20$	2
Deconv3	$32 \times 32 \times 10$	$5 \times 5 \times 20 \times 10$	2
Out(Deconv4)	$64 \times 64 \times 3$	$5 \times 5 \times 10 \times 3$	2

Table 3. Geometric generator's Network Structure.

Layers	In-Out Shape	Conv-Kernel size	stride
Z_g	64×1	Null	Null
Fc1	$4 \times 4 \times 128$	64×2048	Null
Deconv1	$8 \times 8 \times 64$	$3 \times 3 \times 128 \times 64$	2
Deconv2	$16 \times 16 \times 32$	$3 \times 3 \times 64 \times 32$	2
Deconv3	$32 \times 32 \times 16$	$5 \times 5 \times 32 \times 16$	2
Out(Deconv4)	$64 \times 64 \times 2$	$5 \times 5 \times 16 \times 2$	2