Unsupervised Disentangling of Appearance and Geometry by Deformable Generator Network

Xianglei Xing[1], Tian Han[2], Ruiqi Gao[2], Song-Chun Zhu[2], Ying Nian Wu[2]
[1]College of Automation, Harbin Engineering University, Harbin 150001, China
[2]Department of statistics, University of California, Los Angeles, California 90095
xingxl@hrbeu.edu.cn, {hantian,ruiqigao}@ucla.edu,{sczhu,ywu}@stat.ucla.edu

Abstract

We present a deformable generator model to disentangle the appearance and geometric information in purely unsupervised manner. The appearance generator models the appearance related information, including color, illumination, identity or category, of an image, while the geometric generator performs geometric related warping, such as rotation and stretching, through generating displacement of the coordinate of each pixel to obtain the final image. Two generators act upon independent latent factors to extract disentangled appearance and geometric information from images. The proposed scheme is general and can be easily integrated into different generative models. An extensive set of qualitative and quantitative experiments shows that the appearance and geometric information can be well disentangled, and the learned geometric generator can be conveniently transferred to other image datasets to facilitate knowledge transfer tasks.

1. Introduction

Learning disentangled structures behind the observations [2, 26] is a fundamental problem towards understanding and controlling modern deep models. Such disentangled representations are useful not only in building more transparent and interpretable deep models, but also in a large variety of downstream AI tasks such as transfer learning and zero-shot inference where humans excel but machines struggle [22].

Among others, deep generative models, e.g., generator model, have shown great promise in learning representation of images in recent years. However, the learned representation is often entangled and not interpretable. Learning disentangled and interpretable representation for deep generative models is challenging, e.g., from face images where no facial landmarks are given. However, only limited work has been done in this direction.

In this paper, we propose to learn deformable generator model that can disentangle the appearance and geometric information in purely unsupervised manner under a unified probabilistic framework. Specifically, our model integrates two generator networks: one appearance generator and one geometric generator with two sets of independent latent factors. The dense local displacements are generated by the geometric generator, which then act on the image intensities generated by the appearance generator to obtain the final image through a differentiable warping function. The model is learned by introducing alternating back-propagation for two latent factors, and it can also be easily extended to other generative models such as deformable variational auto-encoder. The proposed method can learn well-disentangled representation, which can transfer the appearance and geometric knowledge to other datasets and tasks.

Our contributions are summarized below:

- Propose a deformable generator network to disentangle the appearance and geometric information in purely unsupervised manner.
- The proposed method is general and agnostic. It can be easily extended to different models, such as deformable variational auto-encoder.
- Perform extensive experiments both qualitatively and quantitatively to show that appearance and geometric information can be well disentangled, which can be effectively transferred to other datasets and tasks.

2. Related work

Existing work on learning disentangled representation using deep generative models generally fall into two categories: implicit learning and explicit learning.

The implicit learning methods proceed through latent factors disentanglement and are focused on two categories: the Generative Adversarial Networks (GANs) [10, 8, 28, 23, 33] and the Variational Auto-encoders (VAEs) [17, 30, 27, 21]. InfoGAN [5] and β-VAE [13] are representatives for the two families. Though implicit methods can be learned unsupervisely, their learned representation is not controllable and not well separated.

The explicit methods, on the other hand, model appearance and geometric separately and find their roots in the Active Appearance Models (AAM) which [6, 19] separately learn the appearance and geometric information. Recently, [18] incorporates the shape geometry into the GANs and learns well separated appearance and geometric information. However, these methods [19, 18] require annotated facial landmarks for each image during training. Unsupervised disentangling of the appearance and geometric information is challenging and remains largely unexplored. An independent work proposed recently by [31] follow this direction, but their model is focused on the auto-encoder (AE) only and is not developed under probabilistic framework compared to ours.

3. Model and learning algorithm

3.1. Model

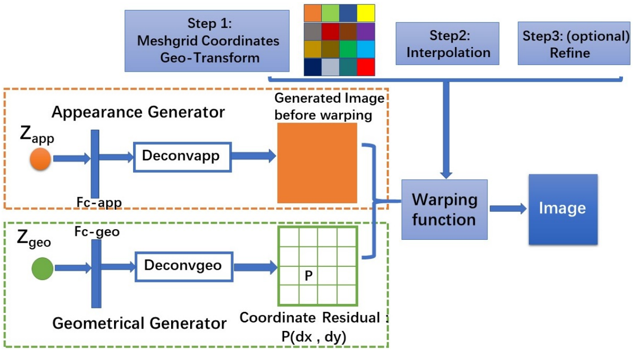


Figure 1. An illustration of the proposed model. The model contains two generator networks: one appearance generator and one geometric generator. The two generators are combined by a warping function to produce the final image. The warping function includes a geometric transformation operation for image 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, which are combined by 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 the appearance, and $Z^g \in \mathbb{R}^{d_g}$ which controls the geometric information. Varying the geometric latent vector Z^g and fixing the appearance latent vector Z^a, we can transform an object's geometric information, such as rotating it with certain angle and changing its shape. On the other hand, varying Z^a and fixing Z^g, we can change the identity or the category of the object, while keeping it geometric information unchanged, such as the same viewing angle or the same shape.

The model can be expressed as

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

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 uses the displacements generated by the geometric generator $F_g(Z^g;\theta_g)$ to warp the image generated by the appearance generator $F_a(Z^a;\theta_a)$ to synthesize the final output image X.

3.2. 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 target coordinate (x, y) for every location (u, v) in the source coordinate. The geometric operation only modifies the positions of pixels in an image without changing the color or illumination. Therefore, the appearance information and the geometric information are naturally disentangled by the two generators in the proposed model.

The geometric transformation Φ can be a rigid affine mapping, as used in the spatial transformer networks [16], or a non-rigid deformable mapping, which is the case in our work. Specifically, the coordinate displacement (dx, dy) (or the dense optical flow field) of each regular grid (x, y) in the output warped image X are generated by the geometric generator $F_g(Z^g;\theta_g)$. The point-wise transformation in this deformable mapping can be formulated as

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

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) do not always have integer coordinates, each pixel's value of the output warped image X can be computed by a differentiable interpolation operation. Let $X_a = F_a(Z^a;\theta_a)$ denote the image generated by the appearance generator. The warping function F_w can be formulated as

$$X(x, y) = F_I(X_a(x + dx, y + dy)), \tag{3}$$

where F_I is the differentiable interpolation function. We use a differentiable bilinear interpolation:

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

where $M(\cdot) = \max(0, \cdot)$. The details of back-propagation through this bilinear interpolation can be found in [16].

The displacement (dx, dy) is used in the deformable convolutional networks [7]. The computation of coordinates displacement (dx, dy) is known as the optical flow estimation [14, 3, 32, 9, 15, 29]. Our work is concerned with modeling and generating the optical flow, in addition to estimating the optical flow.

The displacement (dx, dy) may result from the motion of the objects in the scene, or the change of viewpoint relative to 3D objects in the scene. 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.

3.3. 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 [12]. Specifically, the proposed model can be trained by maximizing the log-likelihood on the training dataset $\{X_i, i = 1, \dots, N\}$,

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

where we integrate out the uncertainties of Z_i^a and Z_i^g in the complete-data log-likelihood to get the observed-data log-likelihood.

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

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

Since the expectation in Eq.(6) is usually analytically intractable, we employ Langevin dynamics to draw samples from the posterior distribution $p(Z_a, Z_g|X;\theta)$ and compute the Monte Carlo average to estimate the expectation term. 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: we fix Z^g and sample Z^a from $p(Z^a|X; Z^g, \theta) \propto p(X, Z^a; Z^g, \theta)$, and then fix Z^a and sample Z^g from $p(Z^g|X; Z^a, \theta) \propto p(X, Z^g; Z^a, \theta)$. At each sampling step, the latent vectors are updated as follows:

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

where t is the number of steps in the Langevin sampling, $\mathcal{E}_t^a$, $\mathcal{E}_t^g$ are independent standard Gaussian noise to prevent the sampling from being trapped in local modes, and δ is the step size. The complete-data log-likelihood can be evaluated by

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

where C_1 and C_2 are normalizing constants. It can be shown that, given sufficient sampling steps, the sampled Z^a and Z^g follow their joint posterior distribution.

Obtaining fair samples from the posterior distribution by MCMC is highly computational consuming. In this paper, we run persistent sampling chains. That is, the MCMC sampling at each iteration starts from the sampled Z^a and Z^g in the previous 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 sampling. The convergence of stochastic gradient descent based on persistent MCMC has been studied by [34].

For each training example X_i, we run the Langevin dynamics following Eq.(7) to get the corresponding posterior samples Z_i^a and Z_i^g. The sample is then used for gradient computation in Eq.(6). More precisely, the gradient of log-likelihood over θ is estimated by Monte Carlo approximation:

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

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.

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 $\{Z_i^a, Z_i^g, i = 1, \dots, N\}$

  Let $t \leftarrow 0$, initialize $\theta$.
  Initialize $\{Z_i^a, Z_i^g, i = 1, \dots, N\}$
  repeat
    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).
    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).
    Let $t \leftarrow t + 1$
  until $t = T$
```

3.4. Deformable Variational Auto-encoder

The proposed deformable generator scheme is general and agnostic to different models. In fact, our method can also be learned by VAE [17] to obtain deformable variational auto-encoder, by utilizing extra inference network to infer (Z^a, Z^g) through re-parametrization. Specifically, we learn another $q(Z^a, Z^g|X; \phi)$ to approximate the intractable posterior $p(Z^a, Z^g|X; \theta)$. The appearance and geometric latent vectors are assumed to be independent Gaussian in the approximated distribution, i.e., $q(Z^a, Z^g|X; \phi) = q(Z^a|X; \phi)q(Z^g|X; \phi)$, where the means and variances are modeled by inference network with parameters ϕ. This deformable VAE model is a naturally extension of the proposed deformable generator framework developed. We show some preliminary results in Sec.4.1.1. Notice that the proposed scheme can also be used in adversarial learning methods[10], by designing a separate discriminator network for shape and appearance. We leave it as our further work. In this work, we focus on the current learning and inference algorithm for the sake of simplicity, so that we do not resort to extra networks.

4. Experiments

In this section, we first qualitatively demonstrate that our proposed deformable generator framework consistently disentangles the appearance and geometric information. Then we analyze the proposed model quantitatively. The structures and parameters of the proposed model are listed in the Appendix. In the following experiments, in each row we visualize the generated samples by varying a certain unit of the latent vectors within the range $[-\gamma, \gamma]$, where we set γ to be 10.

4.1. Qualitative experiments

4.1.1 Experiments on CelebA

We first train the deformable generator on the 10,000 random selected face images from CelebA dataset [24]. Selected images are processed by the OpenFace [1] and further cropped to 64×64 pixels.

To study the performance of the proposed method in 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 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 2. (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 3.

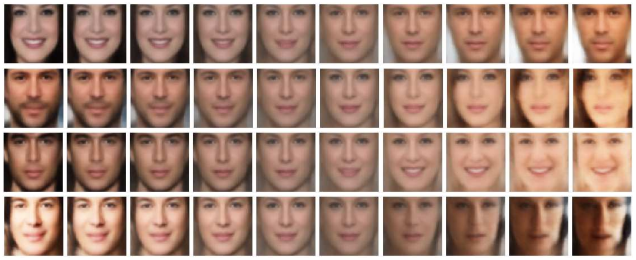


Figure 2. Each dimension of the appearance latent vector encodes appearance information such as color, illumination and gender. In the fist line, the color of background and the gender change. In the second line, the moustache of the man and the hair of the woman vary. In the third line, the skin color changes from dark to white. In the fourth line, the illumination lighting changes from the left-side of the face to the right-side of the face.

As we can observe from figure 2, (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 fist line of figure 2, 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 2, the moustache of 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

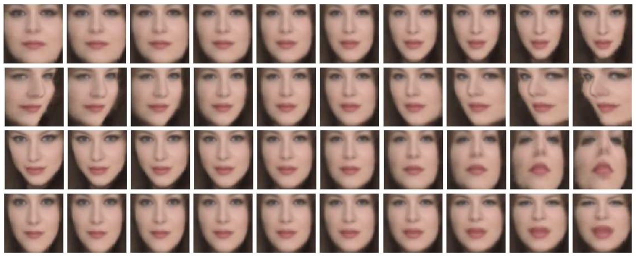


Figure 3. Each dimension of the geometric latent vector encodes fundamental geometric information such as shape and viewing angle. In the fist line, the shape of the face changes from fat to thin from left to the right. In the second line, the pose of the face varies from left to right. In the third 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.

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 3, 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 fist line of figure 3, 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.

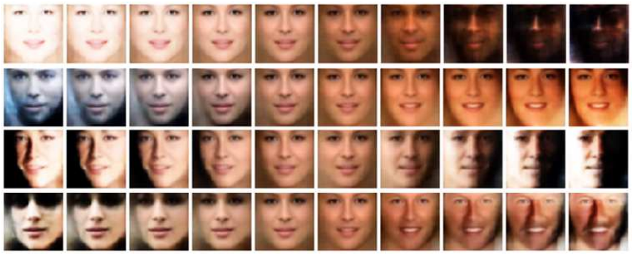


Figure 4. Appearance interpolation results by deformable VAE.

The appearance and geometric information could also be effectively disentangled by the introduced deformable VAE. For the extra inference network, or encoder network, we use the mirror structure of our generator model in which we use convolution layers instead of convolution transpose layers. The generator network structure as well as other parameters are kept the same as the model learned by alternating back-propagation. Figures 4 and 5 show interpolation results following the same protocol described before.

From the results in figures 2 and 3, 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 3) learned by the geometric generator to all the canonical faces (e.g. generated faces in figure 2) learned by the appearance generator. Figure 6 demonstrates the effect of applying geometric warping to the generated canonical faces in figure 2. Comparing figure 2 with figure 6, 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.

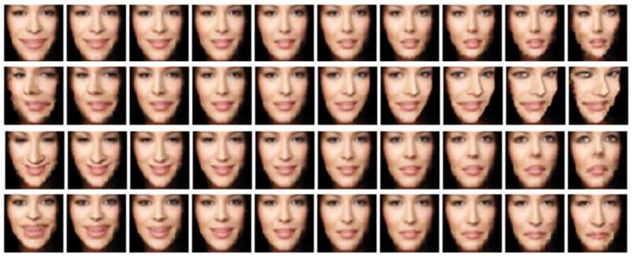
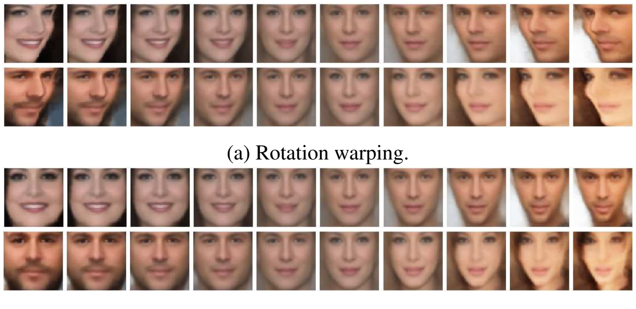


Figure 5. Geometry interpolation results by deformable VAE.



(a) Rotation warping.

(b) Shape warping.

Figure 6. 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 2, only the pose information varies, and the identity information is kept in the process of warping.

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 7. 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 7. 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

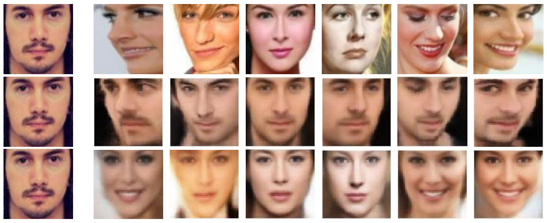


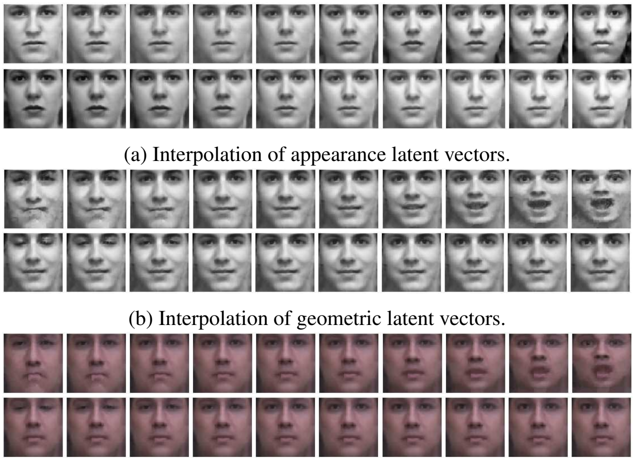
Figure 7. Transferring and recombining geometric and appearance vectors. The first row shows 7 unseen faces from CelebA. The second row shows the generated faces by transferring and recombining 2th-7th faces' geometric vectors with first face's appearance vector in the first row. The third row shows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

we can observe from the second row of figure 7, (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 7 have the same geometric vector Z^g_1, but the appearance vectors Z^a are swapped between each image pair. From the third row of figure 7, 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 model an unseen face accurately.

4.1.2 Experiments on expression dataset

We next study the performance of the proposed deformable generator model on the face expression dataset CK+ [25]. 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. We do not use the labels of expressions provided by CK+ dataset in the learning. 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 [11], 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) Transferring 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) Transferring the learned expression to the face images in Multi-PIE dataset.

4.1.3 Experiment on non-face dataset

We could transfer and learn the model on more general dataset other than face images. For example, the learned geometric information from the CelebA face images can be directly transferred to the faces of animals such as cats and monkeys, as shown in figure 9. The cat faces rotate from left to right and the shape of monkey faces changes from fat to thin, when the warpings learned from human faces are applied.

We also learn our model on the CIFAR-10 [20] dataset, which includes 50,000 training examples of various object categories. 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, i.e., shape and rotation warping.

4.2. Quantitative experiments

4.2.1 Covariance between the latent vectors and geometric variation

First we quantitatively study the covariance between each dimension of the latent vectors (Z^g, Z^a) and input images with geometric variation. We use images with ground-truth labels that record geometric attributes, specifically the multi-view face images from the Multi-Pie dataset [11]. 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 $\theta \in$

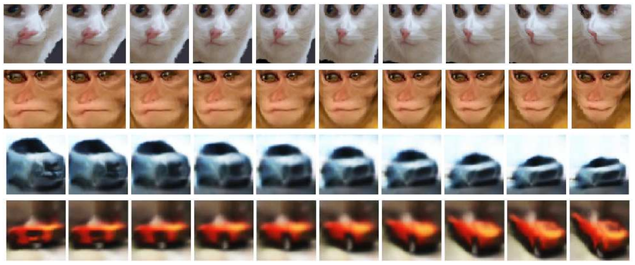


Figure 9. Transferring and learning model from non-face datasets. The first two rows show geometric interpolation results of cat and monkey faces after applying the rotation and shape warping learned from CelebA. The last two rows show geometric interpolation results of the model learned from *car* category of CIFAR-10 dataset.

$\{-30^\circ, -15^\circ, 0^\circ, 15^\circ, 30^\circ\}$, 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 $\bar{Z}^g(i) = [\bar{Z}^g_{-30^\circ}(i), \bar{Z}^g_{-15^\circ}(i), \bar{Z}^g_{0^\circ}(i), \bar{Z}^g_{15^\circ}(i), \bar{Z}^g_{30^\circ}(i)]$. 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. R^g tends to be much larger than R^a.

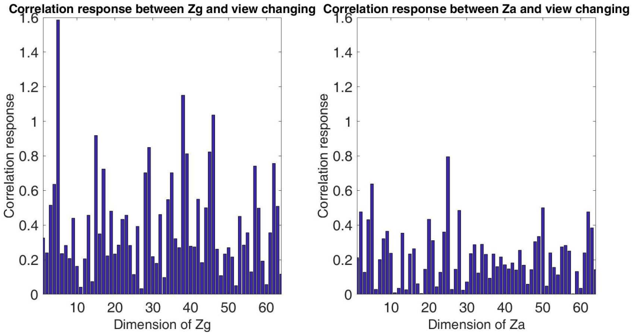
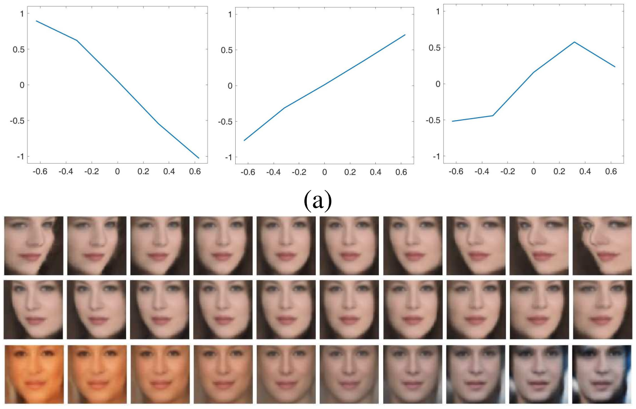


Figure 10. Absolute value of covariance between each dimension of the geometric (or appearance) latent vectors and view variations for the face images from Multi-Pie. The left subfigure shows covariance with the geometric latent vector; the right subfigure shows covariance with the appearance latent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



(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.

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 subfigures from figure 11, the $\bar{Z}^g(i)$ corresponding to the two largest R^g_i (R^g_5, R^g_{38}) is obviously inversely proportional or proportional to the change of viewing angle. However, as shown in the right subfigure, 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.

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, where R^g_5 is inversely proportional to the viewing angle. In the second row, the faces rotate from left to right, where R^g_{38} is proportional to the viewing angle. (3) It is difficult to find obvious variation in viewing angle in the third row. These generated images further verify that the geometric generator of the proposed model mainly captures geometric variation, while the appearance

Methods MSRE	VAE	ABP	Ours
30°	110.99 ± 0.11	117.28 ± 0.12	$\mathbf{89.94 \pm 0.10}$
15°	88.98 ± 0.09	94.81 ± 0.10	$\mathbf{70.64 \pm 0.08}$
0°	48.78 ± 0.05	48.36 ± 0.06	$\mathbf{46.10 \pm 0.06}$
-15°	87.89 ± 0.10	94.12 ± 0.11	$\mathbf{75.11 \pm 0.09}$
-30°	107.94 ± 0.12	120.58 ± 0.13	$\mathbf{92.66 \pm 0.11}$
all views	89.02 ± 0.13	94.66 ± 0.12	$\mathbf{76.52 \pm 0.10}$

Table 1. Comparison of the Mean Square Reconstruction Errors (MSRE) per image (followed by the corresponding standard derivations inside the parentheses) of different methods for unseen multi-view faces from the Multi-Pie dataset.

generator is not sensitive to geometric variation.

4.2.2 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 [11], 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 $\{-30^{\circ}, -15^{\circ}, 0^{\circ}, 15^{\circ}, 30^{\circ}\}$, with 1000 faces for each view. We compare the proposed model with the state-of-art generative models, such as VAE [17, 4] and ABP [12]. For fair comparison, we first train the original non-deformable 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. We perform 10 independent runs and report the mean square reconstruction error per image and standard derivation over the 10 trials for each method under different views as shown in Table 1. Deformable generator network obtains the lowest reconstruction error. When the testing images are from the view closing to that from the training images, all the three methods can obtain small reconstruction errors. When various views of the testing images are included, deformable generator network obtains obviously smaller reconstruction error. Our model benefits from the transferred geometric knowledge learned from the CelebA dataset, while both the non-deformable VAE and ABP models cannot efficiently learn or transfer purely geometric information.

4.3. Balancing explaining-away competition

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. Two generators 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 of the appearance and geometric generators. We tune the α carefully and set it to 0.625 in our experiments.

5. Conclusion

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 beyond the training dataset 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 for disentanglement in general.

The model can be generalized to model dynamic patterns by adding transition models for the latent vectors. The transition model for the appearance vector may generate dynamic textures of non-trackable motion, while 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.

Acknowledgment

This work was done while the first author worked as a visiting scholar at UCLA and supported by Natural Science Foundation of China No. 61703119, Natural Science Fund of Heilongjiang Province of China No. QC2017070 and DARPA XAI project N66001-17-2-4029, ARO project W911NF1810296, and ONR MURI N00014-16-1-2007. We thank Mitchell K. Hill for his assistance with writing.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[20] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

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

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

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

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

[25] Patrick Lucey, Jeffrey F Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. 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*, pages 94–101. IEEE, 2010.

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

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

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

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

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

[31] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming au-

toencoders: Unsupervised disentangling of shape and appearance. *arXiv preprint arXiv:1806.06503*, 2018.

[32] Deqing Sun, Stefan Roth, and Michael J Black. 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.

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

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