Deformable Generator Networks: Unsupervised Disentanglement of Appearance and Geometry

Xianglei Xing, *Member, IEEE,* Ruiqi Gao,Tian Han,Song-Chun Zhu, *Fellow, IEEE,* and Ying Nian Wu

Abstract—We present a deformable generator model to disentangle the appearance and geometric information for both image and video data in a purely unsupervised manner. The appearance generator network models the information related to appearance, including color, illumination, identity or category, while the geometric generator performs geometric warping, such as rotation and stretching, through generating deformation field which is used to warp the generated appearance to obtain the final image or video sequences. Two generators take independent latent vectors as input to disentangle the appearance and geometric information from image or video sequences. For video data, a nonlinear transition model is introduced to both the appearance and geometric generators to capture the dynamics over time. 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.

Index Terms—Unsupervised learning, Deep generative model, Deformable model.

1 Introduction

Learning disentangled structures of the observations [1], [2] is a fundamental problem towards controlling modern deep models and understanding the world. Conceptual understanding requires a disentangled representation that separates the underlying explanatory factors and shows the important attributes of the real-world data explicitly [3], [4]. For instance, given an image dataset of human faces, a disentangled representation can separate the face's appearance attributes, such as color, light source, identity, gender, and the geometric attributes, such as face shape and viewing angle. Such disentangled representations are semantically meaningful not only in building more transparent and interpretable generative models, but also useful for a large variety of downstream AI tasks such as transfer learning and zero-shot inference where humans excel but machines struggle [5]. It has also been shown that such disentangled representations are more generalizable and robust against adversarial attacks [6].

Recently, deep generative models, e.g., generator model, have shown great promise in learning representation of images [7], [8]. Most efforts focus on developing sophisticated architectures and training paradigms for sharp and realistic-looking image synthesis [8], [9], [10]. However, the learned latent representation is often entangled and not interpretable. Learning disentangled and interpretable representation without supervision for deep generative models is challenging, e.g., from face images where no facial landmarks are given. We are particularly interested in learning the disentangled representation for the generative models, since it allows users to specify the desired properties of the output by controlling the generative factors encoded in each latent dimension. Many exciting applications require generative models that can synthesize novel instances while certain key factors are held fixed, for example, generating a face image with desired attributes, such as color, face shape, expression and view (which can be learned and transferred from another person), while keeping the identity fixed. There are increasing demands on such generative models in various domains, such as image manipulation [11], [12], video generation [13], [14] machine learning fairness [15], [16], and drug discovery [17].

In this paper, we propose a deformable generator model that disentangles the appearance and geometric information and is learned in a 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 deformation fields (displacements of the coordinates of each pixel) are generated by the geometric generator, which 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 alternating back-propagation through the model parameters and latent variables of the two networks. The proposed model can be applied to both image and video data. For the spatial-temporal process in video data, the proposed dynamic deformable model introduces nonlinear transition models for the latent vectors. The transition model for the appearance factors can generate dynamic textures which represent non-trackable motion, and the transition model for the geometric factors can generate intuitive physics which represent trackable motion. The pro-

- *X. Xing is with the College of Automation, Harbin Engineering University, Heilongjiang, China, 150001.*
E-mail: xingxl@hrbeu.edu.cn
- *T. Han is with the Computer Science Department, Stevens Institute of Technology, Hoboken, New Jersey, 07030.*
E-mail: than6@stevens.edu
- *R. Gao, S.-C. Zhu and Y. N. Wu are with Department of statistics, University of California, Los Angeles, California 90095.*
E-mail: {ruiqigao,sczhu,ywu}@stat.ucla.edu

Project page:https://andyxingxl.github.io/Deformable-generator/

posed method can learn well-disentangled representation. The learned appearance and geometric representations can be transferred to other datasets and tasks.

Our contributions are summarized below:

- We propose a deformable generator network to disentangle the appearance and geometric information in a purely unsupervised manner.
- We propose a dynamic deformable generator network to disentangle the appearance and geometric information for the spatial-temporal process in video data.
- We perform extensive experiments both qualitatively and quantitatively which show that the appearance and geometric information can be well disentangled and 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 encourage the disentanglement of latent factors by adding regularization terms in the loss function, which fall into two categories: the Generative Adversarial Networks (GANs) [18], [19], [20], [21], [22] and the Variational Auto-encoders (VAEs) [23], [24], [25], [26]. InfoGAN [27] and β-VAE [28] are representatives for the two families respectively, which focus on designing loss functions to encourage the independence of latent factors. More specifically, InfoGAN [27], which belongs to the GANs family, is proposed under the principle of maximization of the mutual information between the observations and a subset of latent factors. However, its disentangling performance is sensitive to the choice of the prior and the number of latent factors. Recently, a contrastive regularization term [29] is introduced to substantially improve the disentanglement capability of the InfoGAN. β-VAE [28], from the VAEs family, learns disentangled representations by utilizing a VAE objective but with a stronger penalty on the discrepancy between the posterior distributions of the latent factors and independent Gaussian priors, making latent factors to be independent as much as possible, thus giving a more robust and stable solution for disentanglement. Deep convolution inverse graphics network (DC-IGN) [30], a variation of the VAE, aims to learn disentangled and interpretable representation for scenes, such as pose, light and texture. To this end, the training is carried out using a set of inactive and active transformations. Recently, a term measuring the total correlation between latent variables is decomposed from the evidence lower bound to refine the β-VAE for learning disentangled representations [31]. Though implicit methods can be learned unsupervisely, the learned representation is not controllable and not well separated.

The explicit methods, on the other hand, model appearance and geometric explicitly by separate models, originated from the Active Appearance Models (AAM) which [32], [33], [34] learn the appearance and geometric information by performing principal component analysis (PCA) on appearance and facial landmarks separately. Unlike the AAM method which requires hand-annotated facial landmarks, our proposed deformable generator model is purely unsupervised and learns from images or videos alone. Recently, [35] incorporates the shape geometry into the GANs which generalizes the linear AAM model to the nonlinear model to learn well separated appearance and geometric information. However, this method [35] also requires annotated facial landmarks for each image during training. Unsupervised disentanglement of the appearance and geometric information is challenging and remains not well-explored. [36] follows this direction, but their model focused on the autoencoder (AE) only, and cannot generate new images with desired attributes by controlling the latent factors. Moreover, it is not developed under probabilistic framework as ours.

3 Model and Learning Algorithm

This section provides the details of the model for 2D-image data and the corresponding learning and inference algorithm. The dynamic deformable model for 3D-video data will be introduced in the next section.

3.1 Model

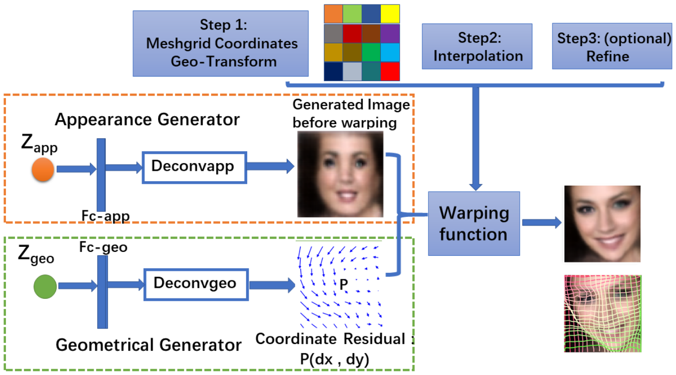


Fig. 1. An illustration of the proposed model. The model contains two generator networks: one appearance generator and one geometric generator. The two generators are combined by a warping function to produce the final image. The warping function includes a geometric transformation operation for image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

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

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 deformation field 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 [37], 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 [37].

The displacement (dx, dy) is used in the deformable convolutional networks [38]. The computation of coordinates displacement (dx, dy) is known as the optical flow estimation [39], [40], [41], [42], [43], [44]. Our work is concerned with modeling and generating the optical flow, in addition to estimating the optical flow.

Notice that the displacement (dx, dy) may also indicates 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 [45]. Specifically, the proposed model can be trained by maximizing the log-likelihood on the training dataset $\{X_i, i = 1, \dots, N\}$,

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

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:

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

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:

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

where τ is the number of steps in the Langevin sampling, $\mathcal{E}_\tau^a$, $\mathcal{E}_\tau^g$ 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

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

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

Algorithm 1 Learning and inference algorithm

```
Input:
  (1) training examples ${X_i \in \mathbb{R}^{D_x \times D_y \times 3}, i = 1, \ldots, N}$
  (2) number of Langevin steps $l$
  (3) number of learning iterations $K$
Output:
  (1) learned parameters $\theta$
  (2) inferred latent vectors $\{Z_i^a, Z_i^g, i = 1, \ldots, N\}$

  1: Let $k \leftarrow 0$, initialize $\theta$.
  2: Initialize $\{Z_i^a, Z_i^g, i = 1, \ldots, N\}$
  repeat
    3: Inference back-propagation: For each $i$, run $l$ steps
    of Langevin dynamics to alternately 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 $\theta_{k+1} \leftarrow \theta_k +
    \eta_k L'(\theta_k)$, with learning rate $\eta_k$, where $L'(\theta_k)$ is computed
    according to Eq.(9).
    5: Let $k \leftarrow k + 1$
  until $k = K$
```

Z^g in the previous iteration. The persistent update results in a chain that is long enough to sample from the posterior distribution, and vastly reduces the computational burden of the MCMC sampling. The convergence of stochastic gradient descent based on persistent MCMC has been studied in [46].

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 samples are then used for computing the gradients over parameters as shown in Eq.(6). More precisely, the gradient of log-likelihood over θ is estimated by Monte Carlo approximation:

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

The whole algorithm iterates through two steps: (1) inferential step which infers the latent vectors by 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.

Besides the above learning and inference method, the proposed model can also be learned by VAE [23] with an extra inference network to infer (Z^a, Z^g), or learned by GAN [18] with an extra discriminator network. In this work, we use the current learning and inference algorithm mainly for the sake of simplicity, so that we do not need to recruit to extra networks.

4 Dynamic deformable model and learning algorithm

For the spatial-temporal process in video, we propose a dynamic deformable model to disentangle the appearance and geometric information. Specifically, a sequence of appearance latent factors $\{Z_t^a, t = 0, \ldots, T\}$, and a sequence of geometric latent factors $\{Z_t^g, = 0, \ldots, T\}$ are fed into two generator networks, which are combined by a warping function to produce the observed video sequence $\mathbf{X} = \{X_0, \ldots, X_T\}$. Suppose $Z_0^a \in \mathbb{R}^{d_a}$ and $Z_0^g \in \mathbb{R}^{d_g}$ are the appearance and geometric latent factors of the first frame, then

$$
\begin{aligned}
Z_t^a &= Z_0^a + s_t^a, \\
Z_t^g &= Z_0^g + s_t^g, \qquad (10)
\end{aligned}
$$

where $s_t^a \in \mathbb{R}^{d_a}$ and $s_t^g \in \mathbb{R}^{d_g}$ are the hidden state vectors that capture the dynamic relation among the video sequence data, $s_0^a = s_0^g = 0$. Inspired by [47] which employs linear auto-regressive model to model the dynamic textures, we introduce non-linear auto-regressive models to model the transition between the hidden state vectors s_t^a and s_t^g:

$$
\begin{aligned}
s_{t+1}^a &= f(s_t^a, \xi_{t+1}^a; \alpha) \\
s_{t+1}^g &= f(s_t^g, \xi_{t+1}^g; \beta) \qquad (11)
\end{aligned}
$$

where ξ_{t+1}^a and ξ_{t+1}^g are independent Gaussian noise vectors, that encode the randomness in the transition from s_t^a to s_{t+1}^a and s_t^g to s_{t+1}^g. $f(\cdot; \alpha)$ and $f(\cdot; \beta)$ are two feedforward neural networks or multi-layer perceptrons, where α and β denote the weight and bias parameters of the two networks.

The dynamic deformable generator model can be expressed as

$$
X_t \quad = F_w(F_a(Z_t^a; \theta_a), F_g(Z_t^g; \theta_g)) + \epsilon_t \qquad (12)
$$

where F_w is the warping function as described in Section 3.2. $F_a(\cdot; \theta_a)$ and $F_g(\cdot; \theta_g)$ are the emission models, more specially, the appearance generator and geometric generator, θ_a and θ_g denote the weight and bias parameters of the two generator networks. $\epsilon_t \sim \mathrm{N}(0, \sigma^2 I_D)$ is the independent residual error.

The proposed dynamic deformable generator model can be learned by alternating back-propagation for two sequences of latent vectors without introducing assisting inference network. Our method is motivated by a maximum likelihood learning algorithm for time series data. Specifically, let $\boldsymbol{\theta} = \{\theta_a, \theta_g, \alpha, \beta\}$ consists of all the network parameters to be learned. Let $\mathbf{Z^a} = \{Z_0^a, \xi_1^a, \ldots, \xi_T^a\}$ denotes the appearance related latent vectors and $\mathbf{Z^g} = \{Z_0^g, \xi_1^g, \ldots, \xi_T^g\}$ be the geometric related latent vectors. Both $\mathbf{Z^a}$ and $\mathbf{Z^g}$ can be inferred from the observed video sequences $\mathbf{X}$. We can formulate

$$
\mathbf{X} \quad = F(\mathbf{Z^a}, \mathbf{Z^g}; \boldsymbol{\theta}) + \boldsymbol{\epsilon} \qquad (13)
$$

where $F(\cdot, \cdot; \boldsymbol{\theta})$ composes F_a, F_g and f over time, and $\boldsymbol{\epsilon} = \{\epsilon_t, t = 0, \ldots, T\}$ denotes the observation errors.

The proposed model can be trained by maximizing the log-likelihood on the training video dataset $\{\mathbf{X_i}, i = 1, \dots, N\}$,

$$\begin{aligned} L(\boldsymbol{\theta}) &= \frac{1}{N}\sum_{i=1}^{N}\log p(\mathbf{X_i};\boldsymbol{\theta}) \\ &= \frac{1}{N}\sum_{i=1}^{N}\log\int p(\mathbf{X_i},\mathbf{Z_i^a},\mathbf{Z_i^g};\boldsymbol{\theta})d\mathbf{Z_i^a}d\mathbf{Z_i^g}, \end{aligned} \quad (14)$$

The gradient of the observed-data log-likelihood $L(\boldsymbol{\theta})$ can be evaluated similarly as in Eq. (6) of Section 3.3,

$$\frac{\partial}{\partial\boldsymbol{\theta}}\log p(\mathbf{X};\boldsymbol{\theta}) = \mathrm{E}_{p(\mathbf{Z^a},\mathbf{Z^g}|\mathbf{X};\boldsymbol{\theta})}\left[\frac{\partial}{\partial\boldsymbol{\theta}}\log p(\mathbf{X},\mathbf{Z^a},\mathbf{Z^g};\boldsymbol{\theta})\right] \quad (15)$$

We employ the Monte Carlo average to approximate the above expectation. Specifically, we sample from the posterior distribution $p(\mathbf{Z^a},\mathbf{Z^g}|\mathbf{X};\boldsymbol{\theta})$ alternately by Langevin dynamics to infer the group of latent vectors $\mathbf{Z^a}$ and $\mathbf{Z^g}$. We fix $\mathbf{Z^g}$ and sample $\mathbf{Z^a}$ from $p(\mathbf{Z^a}|\mathbf{X};\mathbf{Z^g},\boldsymbol{\theta}) \propto p(\mathbf{X},\mathbf{Z^a};\mathbf{Z^g},\boldsymbol{\theta})$, and then fix $\mathbf{Z^a}$ and sample $\mathbf{Z^g}$ from $p(\mathbf{Z^g}|\mathbf{X};\mathbf{Z^a},\boldsymbol{\theta}) \propto p(\mathbf{X},\mathbf{Z^g};\mathbf{Z^a},\boldsymbol{\theta})$. At each sampling step, the group of latent vectors are updated as follows:

$$\begin{aligned} \mathbf{Z^a_{\tau+1}} &= \mathbf{Z^a_\tau} + \frac{\delta^2}{2}\frac{\partial}{\partial\mathbf{Z^a}}\log p(\mathbf{X},\mathbf{Z^a_\tau};\mathbf{Z^g_\tau},\boldsymbol{\theta}) + \delta\boldsymbol{\mathcal{E}^a_\tau} \\ \mathbf{Z^g_{\tau+1}} &= \mathbf{Z^g_\tau} + \frac{\delta^2}{2}\frac{\partial}{\partial\mathbf{Z^g}}\log p(\mathbf{X},\mathbf{Z^g_\tau};\mathbf{Z^a_\tau},\boldsymbol{\theta}) + \delta\boldsymbol{\mathcal{E}^g_\tau} \end{aligned} \quad (16)$$

where τ is the number of steps in the Langevin sampling (not to be confused with the time step of the dynamic model,t), $\boldsymbol{\mathcal{E}^a_\tau}$, $\boldsymbol{\mathcal{E}^g_\tau}$ are independent standard Gaussian noise to prevent the sampling from being trapped in local modes, and δ is the step size. $\mathbf{Z^a_\tau} = \{Z^a_{\tau,0},\xi^a_{\tau,1},\dots,\xi^a_{\tau,T}\}$ and $\mathbf{Z^g_\tau} = \{Z^g_{\tau,0},\xi^g_{\tau,1},\dots,\xi^g_{\tau,T}\}$ denote all the sampled appearance and geometric latent vectors at time step τ.

Let $p(\mathbf{X}|\mathbf{Z^a},\mathbf{Z^g},\boldsymbol{\theta}) \sim \mathrm{N}(F(\mathbf{Z^a},\mathbf{Z^g};\boldsymbol{\theta}),\sigma^2\mathbf{I})$, where $\mathbf{I}$ is the identity matrix whose dimension matches that of $\mathbf{X}$. Let $p(\mathbf{Z^a}) \sim \mathrm{N}(0,\mathbf{I_a})$ and $p(\mathbf{Z^g}) \sim \mathrm{N}(0,\mathbf{I_g})$be the prior distribution of $\mathbf{Z^a}$ and $\mathbf{Z^g}$, $\mathbf{I_a}$ and $\mathbf{I_g}$ are the identity matrices whose dimensions match that of $\mathbf{Z^a}$ and $\mathbf{Z^g}$. The complete-data log-likelihood can be evaluated by (assuming $\sigma^2 = 1$, and up to an additive constant)

$$\begin{aligned} &\log p(\mathbf{X},\mathbf{Z^a};\mathbf{Z^g},\boldsymbol{\theta}) = \log\left[p(\mathbf{Z^a})p(\mathbf{X}|\mathbf{Z^a},\mathbf{Z^g},\boldsymbol{\theta})\right] \\ &= -\frac{1}{2}(\|\mathbf{X} - F(\mathbf{Z^a},\mathbf{Z^g};\boldsymbol{\theta})\|^2 + \|\mathbf{Z^a}\|^2) \\ &= -\frac{1}{2}(\sum_{t=0}^{T}\|X_t - F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g))\|^2 + \\ &\qquad \|Z^a_0\|^2 + \sum_{t=1}^{T}\|\xi^a_t\|^2) \\ &\log p(\mathbf{X},\mathbf{Z^g};\mathbf{Z^a},\boldsymbol{\theta}) = \log\left[p(\mathbf{Z^g})p(\mathbf{X}|\mathbf{Z^a},\mathbf{Z^g},\boldsymbol{\theta})\right] \\ &= -\frac{1}{2}(\|\mathbf{X} - F(\mathbf{Z^a},\mathbf{Z^g};\boldsymbol{\theta})\|^2 + \|\mathbf{Z^g}\|^2) \\ &= -\frac{1}{2}(\sum_{t=0}^{T}\|X_t - F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g))\|^2 + \\ &\qquad \|Z^g_0\|^2 + \sum_{t=1}^{T}\|\xi^g_t\|^2) \end{aligned} \quad (17)$$

To infer the detailed components in $\mathbf{Z^a} = \{Z^a_0,\xi^a_1,\dots,\xi^a_T\}$ and $\mathbf{Z^g} = \{Z^g_0,\xi^g_1,\dots,\xi^g_T\}$, for any fixed time point t_0,we have

$$\begin{aligned} &\frac{\partial}{\partial\xi^a_{t_0}}\log p(\mathbf{X},\mathbf{Z^a_\tau};\mathbf{Z^g_\tau},\boldsymbol{\theta}) \\ &= \sum_{t=t_0}^{T}(X_t - F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g)))\times \\ &\quad \frac{\partial F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g))}{\partial F_a(Z^a_t;\theta_a)}\frac{\partial F_a(Z^a_t;\theta_a)}{\partial Z^a_t}\frac{\partial s^a_t}{\partial\xi^a_{t_0}} - \xi^a_{t_0} \\ &\frac{\partial}{\partial Z^a_0}\log p(\mathbf{X},\mathbf{Z^a_\tau};\mathbf{Z^g_\tau},\boldsymbol{\theta}) \\ &= \sum_{t=0}^{T}(X_t - F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g)))\times \\ &\quad \frac{\partial F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g))}{\partial F_a(Z^a_t;\theta_a)}\frac{\partial F_a(Z^a_t;\theta_a)}{\partial Z^a_t} - Z^a_0 \\ &\frac{\partial}{\partial\xi^g_{t_0}}\log p(\mathbf{X},\mathbf{Z^g_\tau};\mathbf{Z^a_\tau},\boldsymbol{\theta}) \\ &= \sum_{t=t_0}^{T}(X_t - F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g)))\times \\ &\quad \frac{\partial F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g))}{\partial F_g(Z^g_t;\theta_g)}\frac{\partial F_g(Z^g_t;\theta_g)}{\partial Z^g_t}\frac{\partial s^g_t}{\partial\xi^g_{t_0}} - \xi^g_{t_0} \\ &\frac{\partial}{\partial Z^g_0}\log p(\mathbf{X},\mathbf{Z^g_\tau};\mathbf{Z^a_\tau},\boldsymbol{\theta}) \\ &= \sum_{t=0}^{T}(X_t - F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g)))\times \\ &\quad \frac{\partial F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g))}{\partial F_g(Z^g_t;\theta_a)}\frac{\partial F_g(Z^g_t;\theta_a)}{\partial Z^g_t} - Z^g_0 \end{aligned} \quad (18)$$

where $\frac{\partial s^a_t}{\partial\xi^a_{t_0}}$ and $\frac{\partial s^g_t}{\partial\xi^g_{t_0}}$ can be computed recursively.

The learning algorithm iterates through two steps: (1) inference step: infers the two group of latent vectors $\mathbf{Z^a}$ and $\mathbf{Z^g}$ through the Langevin dynamics, when the current $\boldsymbol{\theta}$ is given, according to Eq.(16$\sim$ 18). (2) Learning step: update the network parameters $\boldsymbol{\theta}$ by stochastic gradient descent, when the group of samples $\mathbf{Z^a}$ and $\mathbf{Z^g}$ are given, according to Eq. (14) and (15), more precisely,

$$\begin{aligned} &\frac{\partial}{\partial\boldsymbol{\theta}}L(\boldsymbol{\theta}) \approx \frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial\boldsymbol{\theta}}\log p(\mathbf{X_i},\mathbf{Z_i^a},\mathbf{Z_i^g};\boldsymbol{\theta}) \\ &= \frac{1}{N}\sum_{i=1}^{N}\frac{1}{\sigma^2}(\mathbf{X_i} - F(\mathbf{Z_i^a},\mathbf{Z_i^g};\boldsymbol{\theta}))\frac{\partial}{\partial\boldsymbol{\theta}}F(\mathbf{Z_i^a},\mathbf{Z_i^g};\boldsymbol{\theta}). \end{aligned} \quad (19)$$

where $\boldsymbol{\theta} = \{\theta_a,\theta_g,\alpha,\beta\}$, i indexes the video sequence in the training set. For the i-th video sequence, the derivative with respect to the components of $\boldsymbol{\theta}$ are (for convenience and simplicity, omit the index i, assuming $\sigma^2 = 1$)

$$\begin{aligned} \frac{\partial L}{\partial\theta_a} &= \sum_{t=0}^{T}(X_t - F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g)))\times \\ &\quad \frac{\partial F_w(F_a(Z^a_t;\theta_a),F_g(Z^g_t;\theta_g))}{\partial F_a(Z^a_t;\theta_a)}\frac{\partial F_a(Z^a_t;\theta_a)}{\partial\theta_a} \end{aligned}$$

$$\frac{\partial L}{\partial \theta_g} = \sum_{t=0}^{T}(X_t - F_w(F_a(Z_t^a;\theta_a), F_g(Z_t^g;\theta_g)))\times \frac{\partial F_w(F_a(Z_t^a;\theta_a), F_g(Z_t^g;\theta_g))}{\partial F_g(Z_t^g;\theta_g)} \frac{\partial F_g(Z_t^g;\theta_g)}{\partial \theta_g}$$

$$\frac{\partial L}{\partial \alpha} = \sum_{t=0}^{T}(X_t - F_w(F_a(Z_t^a;\theta_a), F_g(Z_t^g;\theta_g)))\times \frac{\partial F_w(F_a(Z_t^a;\theta_a), F_g(Z_t^g;\theta_g))}{\partial F_a(Z_t^a;\theta_a)} \frac{\partial F_a(Z_t^a;\theta_a)}{\partial Z_t^a} \frac{\partial s_t^a}{\partial \alpha}$$

$$\frac{\partial L}{\partial \beta} = \sum_{t=0}^{T}(X_t - F_w(F_a(Z_t^a;\theta_a), F_g(Z_t^g;\theta_g)))\times \frac{\partial F_w(F_a(Z_t^a;\theta_a), F_g(Z_t^g;\theta_g))}{\partial F_g(Z_t^g;\theta_g)} \frac{\partial F_g(Z_t^g;\theta_g)}{\partial Z_t^g} \frac{\partial s_t^g}{\partial \beta} \quad (20)$$

where $\frac{\partial s_t^a}{\partial \alpha}$ and $\frac{\partial s_t^g}{\partial \beta}$ can again be computed recursively. The gradient computations in both the inferential step and the learning step are powered by back-propagation through time (BPTT). Algorithm 2 describes the details of the learning and inference algorithm powered by BPTT for the video sequences.

It is worth to note that, although the variational inference is convenient to be employed for learning a generative model, for this dynamic deformable model, it is difficult to design the inference models to infer the sequence of the group of latent vectors $\mathbf{Z^a} = \{Z_0^a, \xi_1^a, \ldots, \xi_T^a\}$ and $\mathbf{Z^g} = \{Z_0^g, \xi_1^g, \ldots, \xi_T^g\}$ from the video sequence $\mathbf{X} = \{X_0, \ldots, X_T\}$. In contrast, the proposed learning and inference algorithm does not need to design and learn such an extra inference model, and is easy to be implemented. Specifically, we directly sample from the posterior distribution $p(\mathbf{Z^a}, \mathbf{Z^g}|\mathbf{X};\boldsymbol{\theta})$ to implement the inference step in our dynamic deformable model, which is powered by BPTT. Moreover, the proposed learning method directly aims at maximum likelihood, while variational inference targets at maximizing a lower bound.

Algorithm 2 Learning and inference algorithm for video sequences

```
Input:
  (1) training examples $\{\mathbf{X_i} \in \mathbb{R}^{D_x \times D_y \times 3 \times (T+1)}, i = 1, \ldots, N\}$
  (2) number of Langevin steps $l$
  (3) number of learning iterations $K$
Output:
  (1) learned parameters $\boldsymbol{\theta} = \{\theta_a, \theta_g, \alpha, \beta\}$
  (2) inferred group of latent vectors,
  $\mathbf{Z_i^a} = \{Z_{0,i}^g, \xi_{1,i}^a, \ldots, \xi_{T,i}^a\}$ and $\mathbf{Z_i^g} = \{Z_{0,i}^g, \xi_{1,i}^g, \ldots, \xi_{T,i}^g\}$

  1: Let $k \leftarrow 0$, initialize $\boldsymbol{\theta} = \{\theta_a, \theta_g, \alpha, \beta\}$.
  2: Initialize $\{\mathbf{Z_i^a}, \mathbf{Z_i^g}, i = 1, \ldots, N\}$
  repeat
    3: Inference back-propagation through time: For each $i$, run $l$ steps of Langevin dynamics to alternately sample $\mathbf{Z_i^a}$ from $p(\mathbf{Z_i^a}|\mathbf{X};\mathbf{Z_i^g},\boldsymbol{\theta})$, while fixing $\mathbf{Z_i^g}$; and sample $\mathbf{Z_i^g}$ from $p(\mathbf{Z_i^g}|\mathbf{X};\mathbf{Z_i^a},\boldsymbol{\theta})$, while fixing $\mathbf{Z_i^a}$. Starting from the current $\mathbf{Z_i^a}$ and $\mathbf{Z_i^g}$, each step follows Eq.(16∼18).
    4: Learning back-propagation through time: Update $\boldsymbol{\theta^{k+1}} \leftarrow \boldsymbol{\theta^k} + \eta_k L'(\boldsymbol{\theta^k})$, with learning rate $\eta_k$, where $L'(\boldsymbol{\theta^k} = \{\theta_a^k, \theta_g^k, \alpha^k, \beta^k\})$ is computed according to Eq.(19,20).
    5: Let $k \leftarrow k + 1$
  until $k = K$
```

5 Experiments

In this section, we design 5 groups of experiments to demonstrate that our proposed deformable generator framework consistently disentangles the appearance and geometric information. The parameters and architectures of the deformable generator network are summarized in subsection 5.6. In the following experiments, in each row we visualize the generated samples by varying a certain unit of the latent factors within the range $[-\gamma, \gamma]$, where we set γ to be 10. The code and results can be found at the project page [1]

5.1 Experiment 1: Learn the disentangled basis functions for appearance and geometry

To study the performance of the proposed method in disentangling the appearance and geometric information, we first investigate the appearance basis functions and the geometric basis functions of the learned model. We train the deformable generator on 10,000 face images randomly sampled from CelebA dataset [48]. Some examples in CelebA are shown in Figure 2, which are processed by the OpenFace [49] and further cropped to 64×64 pixels. To better understand how our model works, we show the output of the appearance generator overlaid by a canonical grid in the second row of figure 2. The canonical faces in the front view are learned by the appearance generator. By warping the output of the appearance generator with the deformation fields generated by the geometric generator, we obtain the final reconstructed images, which are shown in the third row of figure 2. The deformation fields, which are the output of the geometric generator, are illustrated by the deformed grids overlaid on the reconstruct images.

The appearance and the geometric latent factors can be interpreted as the projection or reconstruction coefficients along the direction of the corresponding appearance and geometric basis functions. The appearance basis function of the learned model can be demonstrated by the generated images from the combinations of the appearance latent factors Z^a and the geometric latent factors Z^g as follows: (1) set the geometric latent factor Z^g to zero, and (2) vary one dimension of the appearance variable Z^a from $[-\gamma, \gamma]$ with a uniform step $\frac{2\gamma}{10}$, while set the other dimensions of Z^a to zero. Some generated images are shown in figure 3. Similarly, the geometric basis function of the learned model can be demonstrated as follows: (1) set Z^a to be a fixed value, and (2) each time vary one dimension of the geometric latent factor Z^g from $[-\gamma, \gamma]$ with a uniform step $\frac{2\gamma}{10}$, while keeping the other dimensions of Z^g at zero. Some generated results are shown in figure 4.

As we can observe from figure 3, (1) although the train-

[1] https://andyxingxl.github.io/Deformable-generator/

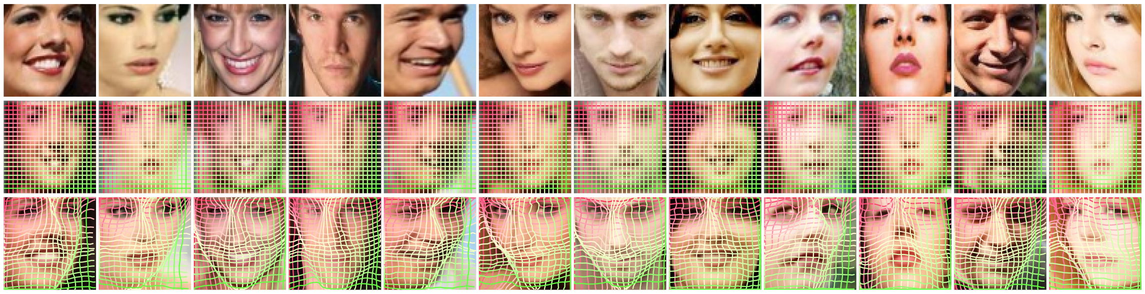


Fig. 2. Example training images from CelebA are illustrated at the first row. The training set contains 10000 images from CelebA, and they are cropped to 64×64 pixels by the OpenFace. These faces have different colors, illuminations, identities, viewing angles, shapes, and expressions. The second row shows the output of the appearance generator overlapped with the canonical grid. The third row demonstrates the deformation fields which is the output of the geometric generator. The deformation fields are visualized by the deformed grids overlaid on the reconstructed images.

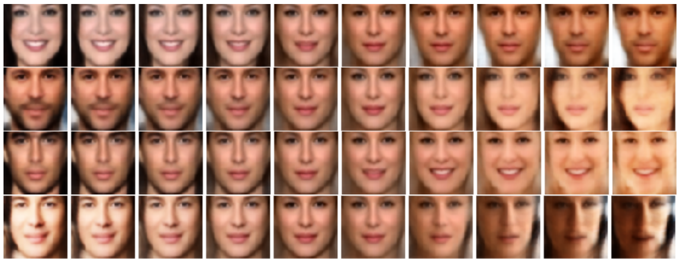


Fig. 3. Typical appearance basis functions, visualized by the generated images that interpolate the appearance latent factors along the basis functions. Each dimension of the appearance latent factors encodes appearance information such as color, illumination and gender. In the fist line, the color of 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.

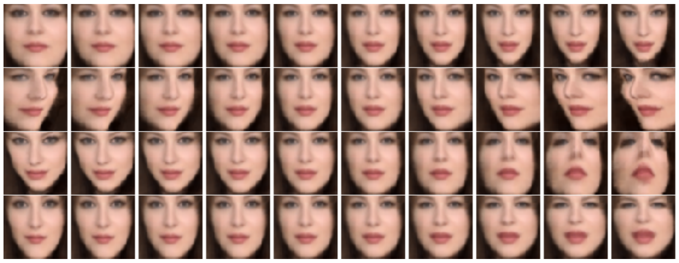


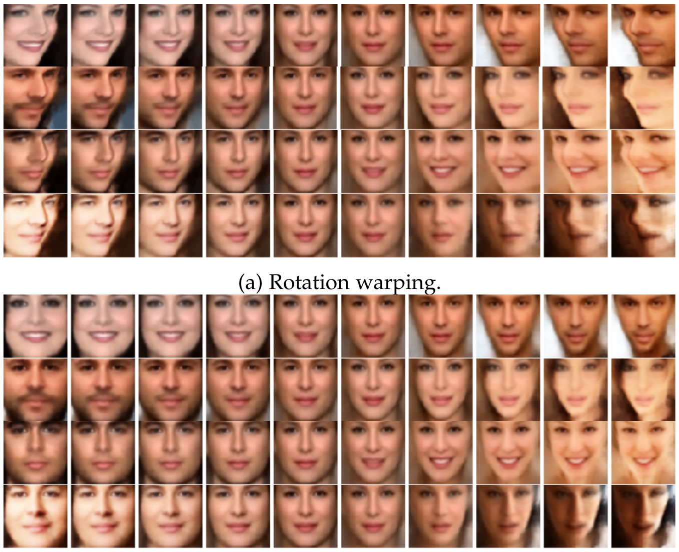
Fig. 4. Representative geometric basis functions, visualized by the generated images that interpolate the geometric latent factors along the basis functions. Each dimension of the geometric latent factors encodes fundamental geometric information such as shape and viewing angle. In the fist line, the 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.

ing faces from CelebA have different viewing angles, the appearance basis functions only encode 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 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 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 basis functions do 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 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. geometric basis functions in figure 4) learned by the geometric generator to all the canonical faces (e.g. appearance basis functions in figure 3) learned by the appearance generator. Figure 5 demonstrates the effect of applying representative geometric basis functions to the appearance basis functions in figure 3. Comparing figure 3 with figure 5, we find that the geometric basis functions which are corresponding to 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.

We next quantitatively study the covariance between the basis functions and input images with geometric variation. We use images with ground-truth geometric attributes, specifically the multi-view face images from the Multi-Pie dataset [50]. The images consist of 5 viewing angles $\{-30°, -15°, 0°, 15°, 30°\}$. For each viewing angle, we randomly sampled 100 images, which are fed into the learned model to infer their geometric latent vector Z^g and appearance latent vector Z^a. For each viewing angle θ , we compute the average $\bar{Z}^g_\theta$ and $\bar{Z}^a_\theta$ of the inferred latent vectors. For



(a) Rotation warping.

(b) Shape warping.

Fig. 5. Applying the geometric basis functions, (a) rotation warping and (b) shape warping, 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.

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 $\bar{Z}^a(i)$ for 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:

$$R^g_i = |\bar{Z}^g(i)^\top \theta|, \quad R^a_i = |\bar{Z}^a(i)^\top \theta| \tag{21}$$

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 R^g and R^a of the geometric and appearance latent vectors in figure 6. R^g tends to be much larger than R^a.

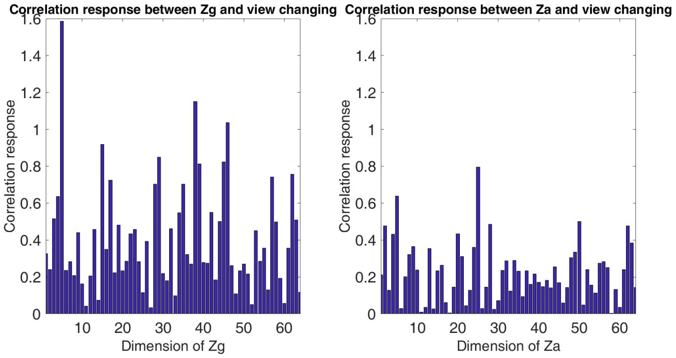
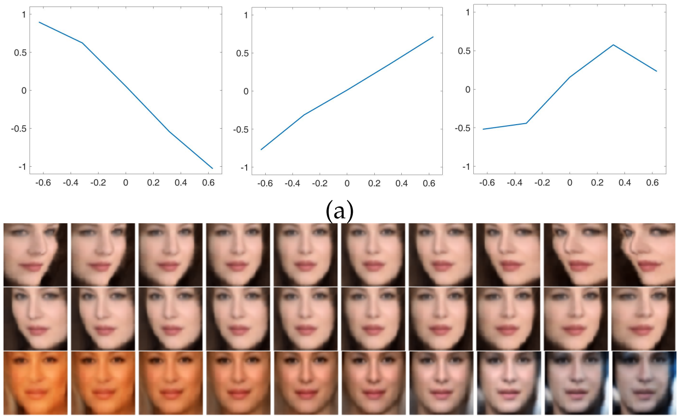


Fig. 6. Absolute value of covariance between each dimension of the geometric (or appearance) latent vectors and view variations for the face images from Multi-Pie. The left subfigure shows covariance with the geometric latent vector; the right subfigure shows covariance with the appearance latent vector.



(b)

Fig. 7. (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 middle) 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 largest R^g_i and 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 7. As we can observe from the left and middle subfigures from figure 7, 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. 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 7(b). We can make several important observations. (1) The variation of viewing angle in the first two rows is very obvious, and variation in the first row is larger than that the one 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 7(a) is steeper than that of the middle subfigure of figure 7(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 generator is not sensitive to geometric variation.

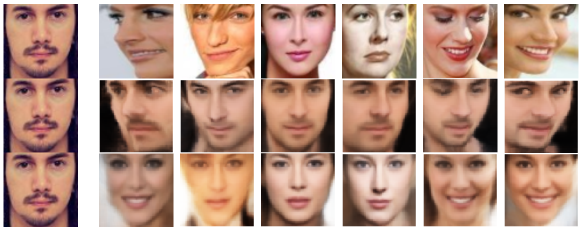


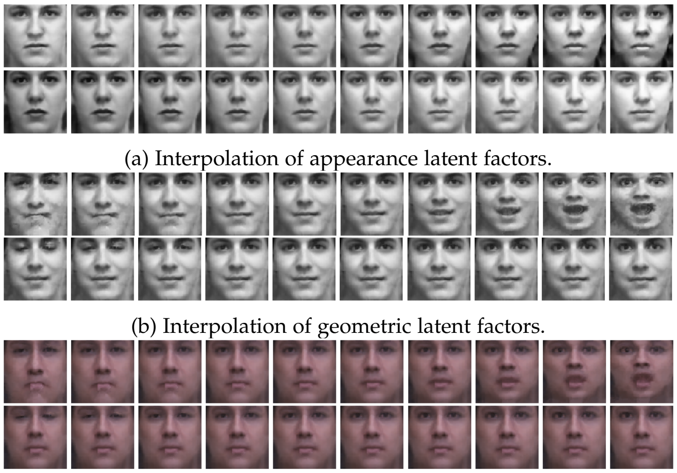
Fig. 8. Transferring and recombining geometric and appearance vectors. The first row shows 7 unseen faces from CelebA. The second row shows the generated faces by transferring and recombining 2th-7th faces' geometric vectors with first face's appearance vector. The third row shows the generated faces by transferring and recombining the 2th-7th faces' appearance vectors with the first face's geometric vector in the first row.

5.2 Experiment 2: Learn to transfer the appearance and geometric knowledge

The disentangling power of the proposed model can be employed to transfer and recombine the geometric and appearance information 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 $\{Z_1^a, Z_2^g\}, \dots, \{Z_1^a, Z_7^g\}$ to generate six new face images, as shown in the second row of figure 8. 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 8. 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 8, (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 8 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 8, 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.

Moreover, we can transfer the learned geometric knowledge from one dataset to another unseen dataset easily. We first train the proposed deformable generator model on the grey face expression dataset CK+ [51]. Following the same experimental protocol as the last subsection, we can investigate the interpolation along the appearance basis functions and the geometric basis functions. The disentangled results are shown in figure 9. 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 basis function usually encodes a neutral expression. The geometric basis function controls major variation in expression, but does not change the identity information.

Then, we try to transfer the learned geometric knowledge, such as expression, from CK+ to another color face dataset, Multi-Pie [50], by fine-turning the appearance generator on the target face dataset while fixing the parameters of the geometric generator. Figure 9 (c) shows the result of transferring the expressions of 9 (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 factors.

(b) Interpolation of geometric latent factors.

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

Fig. 9. Interpolation examples of (a) appearance basis functions and (b) geometric basis functions. (c) Transferring the learned expression to the face images in Multi-PIE dataset.

Furthermore, we quantitatively study the power of the proposed deformable generator model to transfer the geometric knowledge learned from one dataset into another unseen dataset. Specifically, given 1000 front-view faces from the Multi-Pie dataset [50], 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 [3], [23] and ABP [45]. 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 of 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.

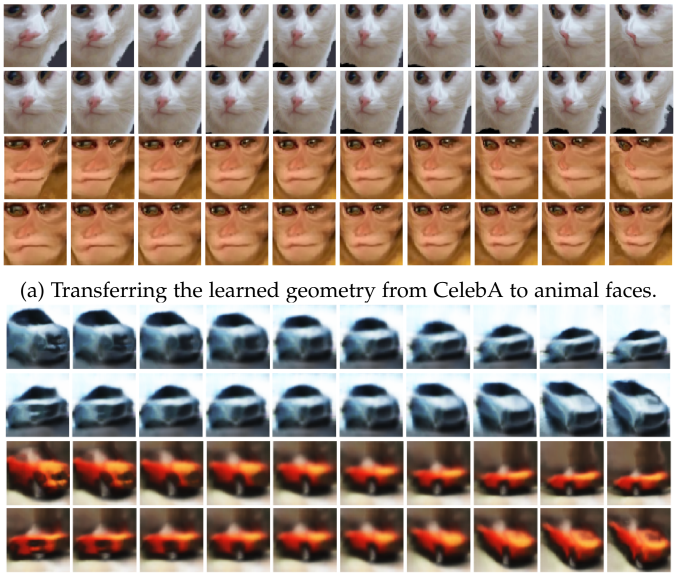
TABLE 1
Comparison of the Mean Square Reconstruction Errors (MSRE) per image (followed by the corresponding standard derivations inside the parentheses) of different methods for unseen multi-view faces from the Multi-Pie dataset.

Methods MSRE	VAE [23]	ABP [45]	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	$\mathbf{76.52 \pm 0.10}$

5.3 Experiment 3: Learn 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 10(a). The cat and monkey faces rotate from left to right and the shape of the animal faces changes from fat to thin, when the warpings learned from human faces are applied.

We also learn our model on the CIFAR-10 [52] dataset, which includes 50,000 training examples of various object categories. To show the result, we randomly sample and fix 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 10(b) shows interpolated examples generated by model learned from the *car* category. The results show that each dimension of Z^g controls a specific geometric transformation, i.e., shape and rotation warping.



(a) Transferring the learned geometry from CelebA to animal faces.

(b) Geometric interpolations that learned from CIFAR-10.

Fig. 10. Transferring and learning model from non-face datasets. (a) Geometric interpolation results of cat and monkey faces after applying the rotation and shape warping learned from CelebA. (b) Geometric interpolation results of the model learned from *car* category of CIFAR-10 dataset.

We next learn our model on the MNIST dataset of 10 classes of digits. In this experiment, Z^a is set to be the prior discrete one-hot label, while Z^g is the continues latent vector. Figure 11 demonstrates the interpolated examples. On each row, we set Z^a to be one of the discrete label, while interpolating one dimension of the geometric latent factor Z^g from $[-\gamma, \gamma]$ with a uniform step $\frac{2\gamma}{10}$. In the left subfigure of 11, the first column represent the images generated by the one-hot Z^a (before warping by the deformable fields generated by Z^g), and the remain 10 columns show the results by interpolating the shape factor of Z^g. As we can observe, from left to right, the shape of the digits change from large to small. Similarly, in the right subfigure of 11, the viewing angle of the digits vary from left to right.

5.4 Experiment 4: Unsupervised landmark localization

In this subsection, we evaluate the performance of deformable generator model on the task of unsupervised face landmark localization. The experimental results are evaluated on the Multi-Attribute Facial Landmark (MAFL) [53] dataset, which contains landmark locations (eyes, nose, and mouth corners) manually annotated for 19k training and 1k test images.

In this experiment, we train our deformable generator model on the CelebA dataset without any supervision. According to the evaluation protocol of the previous work [36], [54], employing the provided training annotations in MAFL, we train a landmark regressor post-hoc on the learned deformation fields from the geometric generator. It is worth to mention that the annotation from the MAFL training set is only employed to train the regressor, while our deformable generator model is trained fully unsupervisedly and fixed. The regressor composes of a MLP with 2-layers. The flattened deformation fields (vectors of size $64 \times 64 \times 2$), learned from the geometric generator, are fed as input to a hidden layer with 100 neurons, followed the ReLU activation function and an output layer with 10 neurons to predict the 2d spatial coordinates (x, y) for the five landmarks locations (eyes, nose, and mouth corners). The L_1 loss is used as the objective function for the landmark regressor.

The first row of figure 12 shows the examples of the testing images. The second row of figure 12 demonstrates the estimated deformed grid. The deformation grid is obtained by warping the canonical grid with the deformation fields learned from the geometric generator. The third row of figure 12 shows the canonical grid overlaid on the canonical faces learned from the appearance generator. The fourth row of figure 12 demonstrates the semantic landmark locations overlapped on the testing images. The green points denote the ground truth, and the red points denote the predictions. As we observe from figure 12, the learned deformation fields could be used for an effective mapping between the landmark locations on the originally unaligned faces and those on the canonical texture faces.

We further quantitatively evaluate the landmark localization by reporting the mean error as a percentage of the inter-ocular distance on the MAFL testing set. We compare our deformable generator model with the other 5 state-of-the-art landmark localization methods [36], [53], [54], [55], and report the results on table 2. As we can observe from table 2, our method outperforms the other state-of-the-art methods, because our method estimate the deformation field more accurately, even though we never explicitly train the deformable generator to learn correspondence.

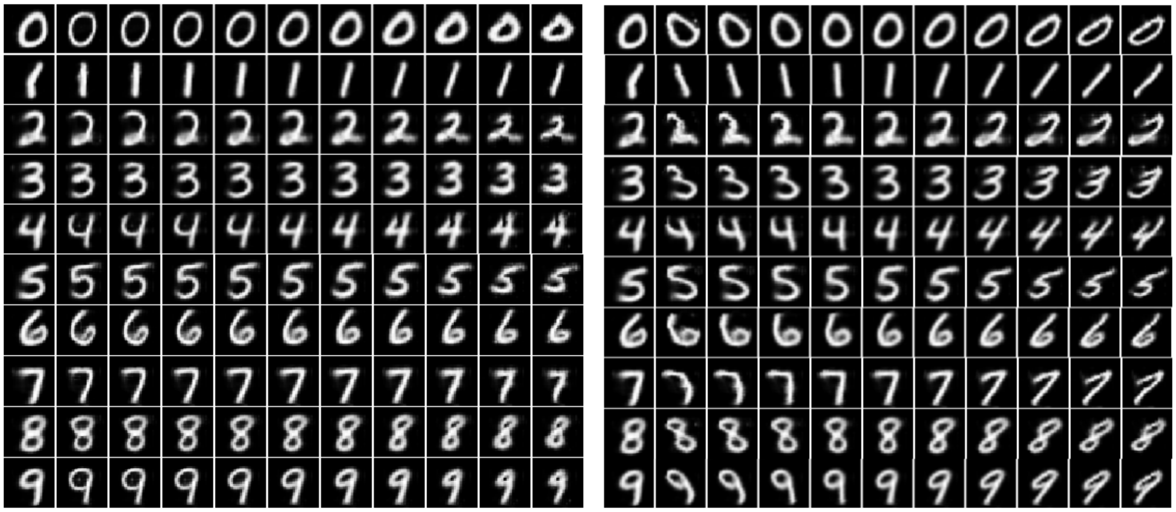


Fig. 11. Interpolation results by the geometric latent factors of the model learned from the MNIST dataset. Each row demonstrates the results of interpolating one dimension of the geometric latent factor, while keeping Z^a to be one discrete label. In the left subfigure, the first column represent the images generated by the one-hot Z^a, and the remain 10 columns show the results by interpolating the shape factor of Z^g. As we can observe, from left to right, the shape of the digits change from large to small. Similarly, as we can observe from the right subfigure, the viewing angle of the digits vary from left to right by interpolating the view factor of Z^g

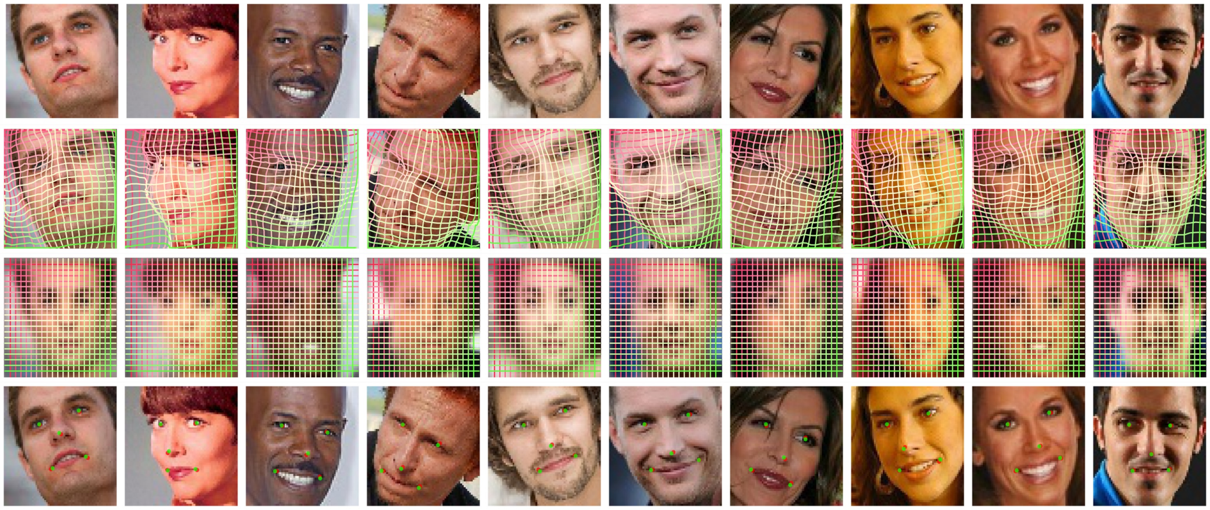


Fig. 12. Unsupervised landmark localization. Row 1: the samples of the testing images from the MAFL dataset. Row 2: the deformation grid estimated from warping the the canonical grid with the coordinate displacement (deformation fields) learned from the geometric generator. Row 3: the canonical grid overlapped on the canonical faces learned from the appearance generator. Row 4: the semantic landmark locations. The green points denote the ground truth, and the red points denote the predictions.

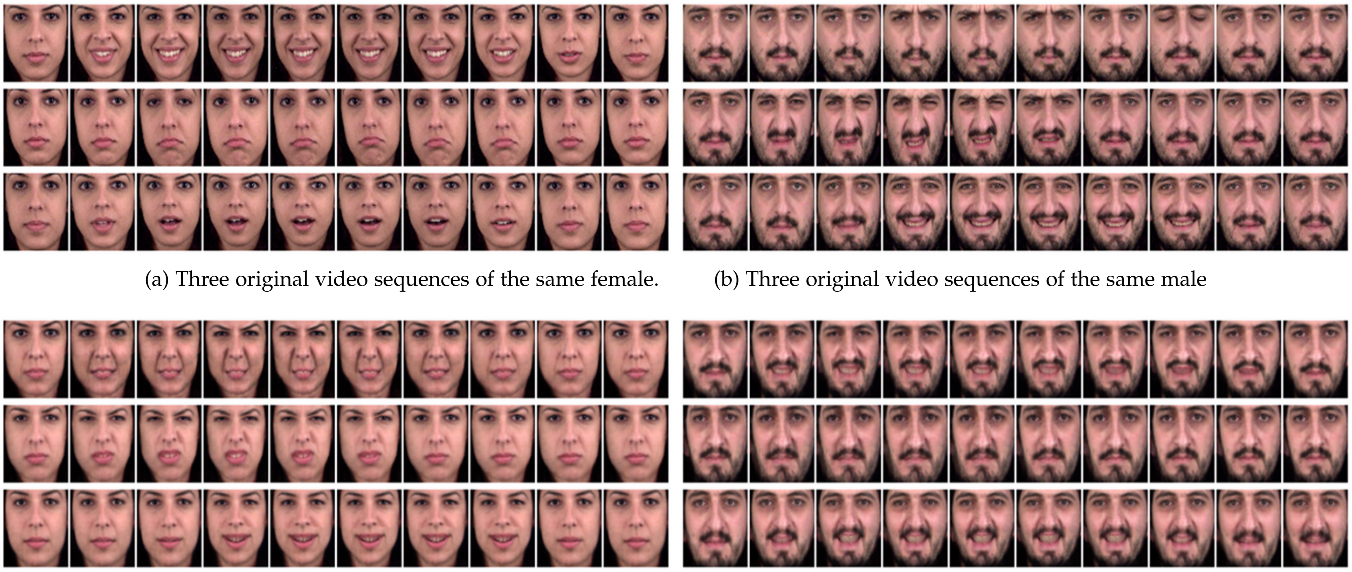
TABLE 2
Comparisons of the mean error of unsupervised landmark prediction on the MAFL test set. Smaller is better.

TCDCN [53]	Thewlis et al. [54]	Dense-DAE [36]	MTCNN [55]	Ours
7.95	5.83	5.45	5.39	**5.18**

5.5 Experiments for Dynamic Deformable Generator

To study the performance of the proposed dynamic deformable generator in disentangling the appearance and geometric information from the video sequences, we experiment on the MUG [56] facial expression video dataset. The dataset consisted of 86 subjects. The video sequences of each subject represent one of the six facial expressions: happiness, sadness, anger, fear, disgust, and surprise. We crop the face regions by the OpenFace and scaled to 64 × 64 pixels ×60 frames. Some example video sequences of MUG are shown in the first 3 rows of figure 13 and figure 14.

To evaluate the performance of disentanglement on the video sequences, we first consider the experiment on transferring and recombining the appearance and geometric information from different video sequences. Specifically, consider two video sequences from different persons. We first learn and infer the sequences of appearance and geometric latent factors from the two videos as $\{Z^a_{1,t}, Z^g_{1,t}, t =$



(a) Three original video sequences of the same female.

(b) Three original video sequences of the same male

(c) Recombine the appearance from (a) and geometry from (b).

(d) Recombine the appearance from (b) and geometry from (a)

Fig. 13. Transfer and recombine the appearance and geometric information from different video sequences. (a) Three original video sequences of the same female with expressions of happiness, sadness and surprise; (b) Three original video sequences of the same male with expressions of anger, disgust, and happiness. (c) Transfer and recombine the appearance from female in (a), and the geometry from the male in (b), so that the generated video sequences in (c) inherit the appearance from the female but the shape and expressions (anger, disgust, and happiness) from the male. (d) Transfer and recombine the appearance from the male in (b), and the geometry from the female in (a), so that the generated video sequences in (d) inherit the appearance from the male but the shape and expressions (happiness, sadness and surprise) from the female.

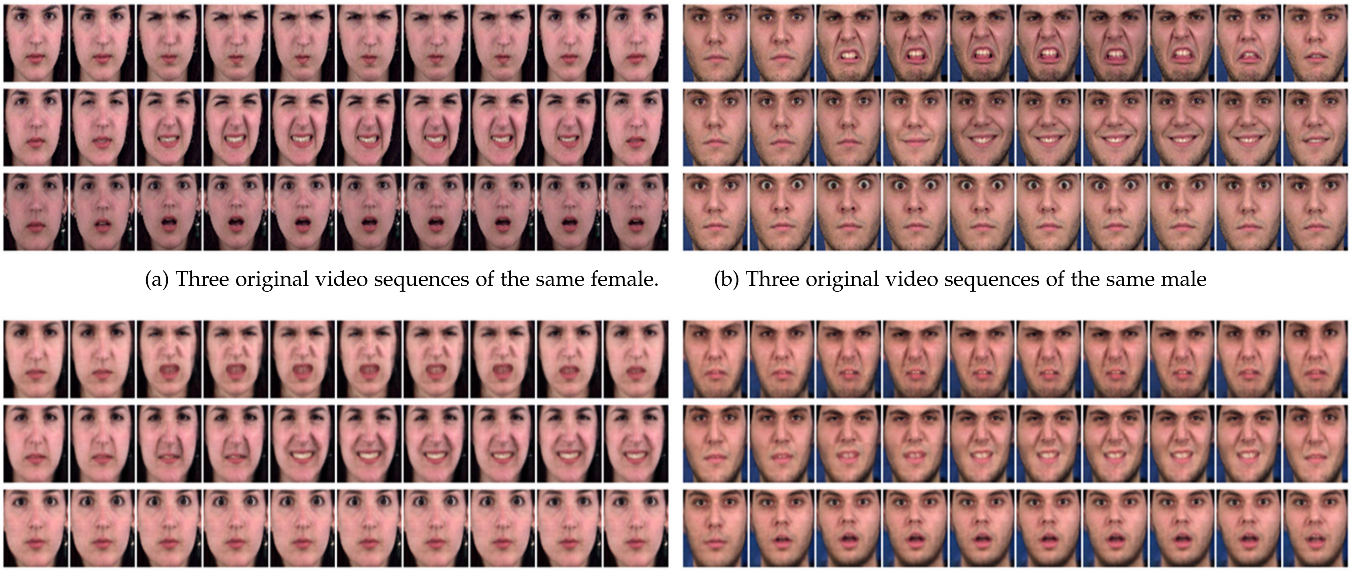
$0, \ldots, T\}$ and $\{Z^a_{2,t}, Z^g_{2,t}, t = 0, \ldots, T\}$. Then we transfer and recombine them to generate new video sequences. More specifically, we expect that recombining the sequence of appearance latent factors from the the first person and the sequence of geometric latent factors from the second person, $\{Z^a_{1,t}, Z^g_{2,t}, t = 0, \ldots, T\}$, will generate a new video that inherits the appearance information from the first person, while inherits the dynamic geometric information (including both the face shape and dynamic face expression) from the second person. Similarly, we also expect that recombining the sequence of appearance latent factors from the the second person and the sequence of geometric latent factors from the first person, $\{Z^a_{2,t}, Z^g_{1,t}, t = 0, \ldots, T\}$, will generate another new video that inherits the appearance information from the second person, while inherits the dynamic geometric information from the first person.

Figure 13 demonstrates the experimental results on transferring and recombining the appearance and geometric information from different video sequences. Figures 13 (a) and (b) show 6 different expressional video sequences of a female and a male. Each row of figure 13 (c) demonstrates a generated video sequence by recombining the sequence of appearance latent factors from the same row of the figure 13 (a) and the sequence of geometric latent factors of the same row from the figure 13 (b). Figure 13 (d) demonstrates the generated videos sequences by recombining the sequence of appearance latent factors from the male and the sequence of geometric latent factors from the female.

As we observe from figure 13 (c), the appearance information, such as the face color, texture, and the five senses, are the same as that of the female, while the geometric information, such as the face shape and dynamic face expression, are similar with the that of the female. In figure 13 (d), the appearance information are the same as that of the male, while the face shape and dynamic face expression are similar with the that of the female. More specifically, comparing figure 13 (d) and figure 13 (b), we can observe that the face shape in figure 13 (d) is obviously thinner than that of the original male's face shape as shown in Subfigure 13 (b), which inherits the face shape of the female from figure 13 (a). Comparing figure 13 (c) and figure 13 (a), we can observe that although the face shape and dynamic face expression are changed, the face color, texture, and the detailed appearance features, such as the eyebrows are kept the same as the original female's eyebrows.

Figure 14 also demonstrates the experimental results on transferring and recombining the appearance and geometric information from video sequences of another two persons. From figure 14, we observe similar phenomena as that from figure 13. The above experimental results verify that the appearance and geometric information from the video sequences are well disentangled by the proposed dynamic deformable generator model.

Next, we further study disentangling of the learned geometric information from the video into the global shape and dynamic expression. Thus, we can edit the face shape of a person, while keeping his dynamic face expression from the original video. Recall in Eq. (10) in Section 4, the sequence of geometric latent factors $Z^g_t, \{t = 0, \ldots, T\}$ can be represented by the summation, in the latent space, of the geometric latent factor of the first frame Z^g_0 and the consequent hidden state vectors $s^g_t, \{t = 1, \ldots, T\}$. Thus, the geometric latent factor of the first frame can be employed to represent the global shape, while the consequent hidden state vectors can be utilized to represent the dynamic expression. We plot the learned appearance basis



(a) Three original video sequences of the same female.

(b) Three original video sequences of the same male

(c) Recombine the appearance from (a) and geometry from (b).

(d) Recombine the appearance from (b) and geometry from (a)

Fig. 14. Transfer and recombine the appearance and geometric information from different video sequences. (a) Three original video sequences of the same female with expressions of anger,disgust and surprise; (b) Three original video sequences of the same male with expressions of disgust, happiness, and fear. (c) Transfer and recombine the appearance from female in (a), and the geometry from the male in (b), so that the generated video sequences in (c) inherit the appearance from the female but the shape and expressions (disgust, happiness, and fear) from the male. (d) Transfer and recombine the appearance from the male in (b), and the geometry from the female in (a), so that the generated video sequences in (d) inherit the appearance from the male but the shape and expressions (anger,disgust and surprise) from the female.

functions for the first frame in the first two rows of figure (15), according to the method we did in Section 5.1. By observing the interpolating results of the appearance latent factors along the basis functions among $[-\gamma, \gamma]$, we find the major appearance basis functions capture the identity and color changing of the person, such as identity varying from male to female (the first row), and the background color varying from black to blue (the second row). Since the MUG dataset only contains face expression videos from the front view, the major geometric basis function captures the shape information. The third and the fourth rows of figure (15) demonstrate the major shape basis function by applying it over the appearance basis functions in the first two rows when keeping the interpolation value equals to $\frac{-4\gamma}{5}$. As we can observe from the interpolation results of the geometric latent factors Z_0^g along the basis functions among $[-\gamma, \gamma]$, the face shape varies from fat to thin, while the identity information from the appearance is kept invariant.

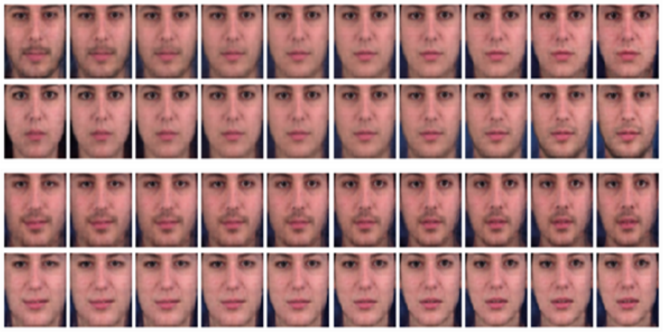


Fig. 15. The first two rows show the typical appearance basis functions for the first frame, visualized by the generated images from interpolating the appearance latent factors along the basis functions. Each dimension of the appearance latent factors encodes appearance information such as gender and background color. The last two rows demonstrate the major geometric basis function, visualized by applying it over the appearance basis functions in the first two rows. The major geometric basis function captures the shape information. From left to the right, the shape of the faces at the last two rows change from fat to thin.

To edit the global shape information of the video sequence, we can recombine the geometric latent factors Z_0^g with different coefficients corresponding to the shape basis function, and the consequent hidden state vectors $s_t^g, \{t = 1, \ldots, T\}$ to generate a new sequence of geometric latent factors $Z_t^g, \{t = 0, \ldots, T\}$. Figure (16) shows the results of editing the shape of two face expression videos. As we can observe from figure (16), the face shapes of the second row and the fourth row are thinner than that of the first and the third row, while the facial expressions, as well as the appearance information, of the second row and the fourth row are the same as that of the first and the third row. These results verify that the proposed dynamic deformable generator model can not only disentangle the appearance and geometric information of the video sequence, but also further disentangle the global shape and dynamical expression among the geometric information.

Dynamic Deformable fields for facial expression analysis and recognition

We next demonstrate that the learned dynamic deformable fields can be used for facial expression analysis and recognition. To visualize the dynamic deformable fields, we plot the dynamic deformed grids over the corresponding facial sequences. As we can observe from figure 17, for different expressions, the dynamic deformed grids are different. The

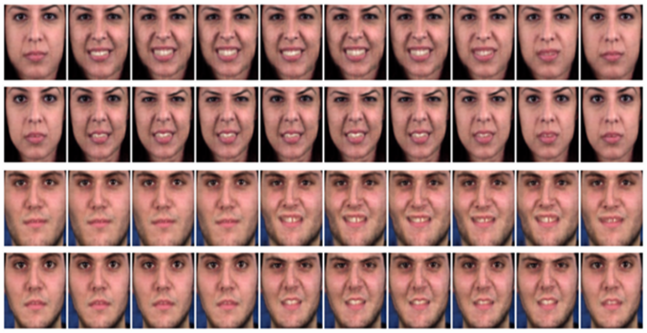


Fig. 16. Editing the shape of the whole video sequence of facial expression. The face shapes of the video sequences in the second and the fourth row are thinner than that of the first and the third row, while the facial expressions, as well as the appearance information, of the video sequences in the second and the fourth row are kept invariant as that of the first and the third row.

first row of figure 17 demonstrate the anger. As observed from the deformable lines around the eyes, brows, and mouth, the implicit movements are inward lowering the brows and mouth compaction. The second row demonstrate the disgust. As observed from the deformable lines around the nose, mouth and brows, the implicit movements are upward nose motion, mouth expanded and opened, and lowering of brows. The third row show the fear. As observed from the deformable lines around the mouth and brows, the implicit movements are slight expansion and raising of mouth and raising inner parts of brows. The fourth row shows the happiness. As observed from the deformed grids around the mouth, the implicit movement is mouth opening with its expansion. The fifth row demonstrates the neutral. As observed from the deformed grids, there are no movements under this neutral expression. The sixth row shows the sadness. As observed from the deformable lines around the mouth, the implicit movement is lowing mouth corners and raising mid mouth. The last row demonstrate the surprise. As observed from the grid lines around the brows and the lip, the implicit movement is raising brows and raising the upper lip.

Since the facial expression is connected with the dynamic geometric information and unrelated with the appearance information, such as color, illumination, and identity, we can employ both the learned dynamic geometric latent factors $\{Z_t^g, t = 0, \cdots, T\}$ and the dynamic deformation fields $\{F_g(Z_t^g; \theta_g), t = 0, \cdots, T\}$ (defined in Eq. 12) as the feature to recognize the facial expressions. Specifically, the learned dynamic geometric latent factors (dglf) or the dynamic deformable fields (ddf) are fed into the long short-term memory (LSTM) to model the temporal change. More specifically, we utilize an one-layer LSTM, and the output of LSTM is connected with a fully connected layer with 30 hidden neurons and a soft-max layer with cross-entropy loss. To evaluate the performance of our method, we compare our method with MoCoGAN [57], whose motion latent factors are feed to the same LSTM and the logistic classifier. We also compare our method with 5 state-of-art facial emotion recognition algorithms: DAGSVM-GT [58], UPSM [59], HiNet [60], Multi-stream CNN [61], and RADAP [62]. According to the experimental setup in RADAP, UPSM and HiNet, in our experiments, the 10-fold person independent cross-validation scheme is employed.

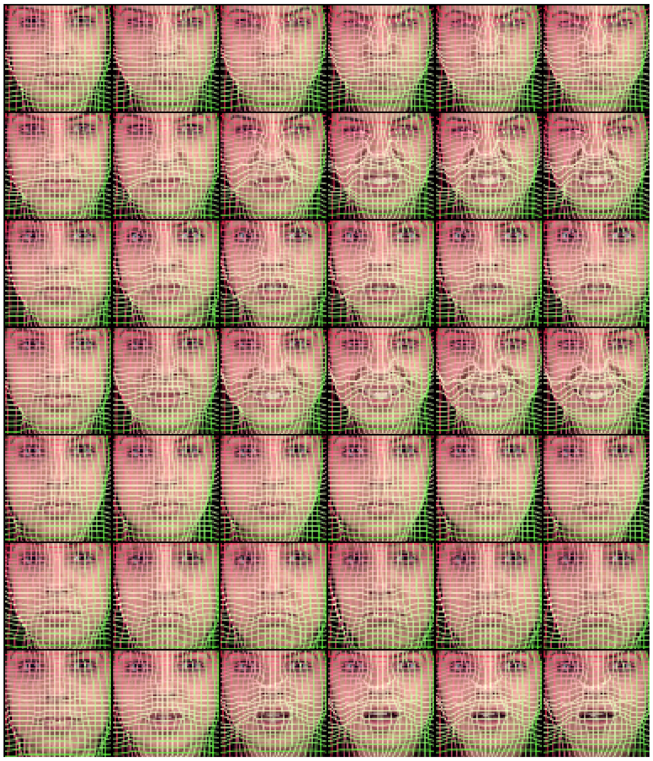


Fig. 17. Dynamic deformed grids for 7 different facial expressions over the corresponding facial sequences. From the top row to the bottom row, the facial expressions are anger, disgust, fear, happiness, neutral, sadness. and surprise. As can be observed from the regions around the eyes, brows, nose, and mouth, the learned dynamic deformation fields (demonstrated as deformed grids) reflect the intrinsic feature of different kinds of facial expressions. See text for detail. Best viewed with magnification.

In table 3, we measure the performance of facial emotion recognition in terms of average recognition accuracy. Our method using dynamic geometric latent factors (dglf) performs better than the MoCoGAN's motion latent factors as the feature representation. Moreover, the learned dynamic deformable fields (ddf) achieves superior recognition accuracy compared with the other algorithms. The reasons are as follows: (1) The disentangled dynamic geometric information is more effective for facial expression recognition, since the appearance information, such as color, illumination, and identity, are unrelated with the expression or the emotion and sometimes may result in negative effects. (2) Our model disentangles the appearance and geometric information by two generators connected by the warping function. The geometric warping only modifies the positions of pixels in an image without changing the color or illumination. Therefore, the extracted dense deformable fields is more effective and pure than the motion latent factors learned by the MoCoGAN, which disentangles the content and motion latent factor from one concatenated latent vector. (3) The learned dense deformable fields contain more important information than the methods based on sparse landmarks such as UPSM [59] and DAGSVM-GT [58].

5.6 Balancing explaining-away competition and Network Structures

The proposed deformable generator model utilizes two generator networks to disentangle the appearance and ge-

TABLE 3
Comparisons of the mean recognition accuracy of seven different methods on the MUG dataset. Dglf and Ddf denote our method employ the learned dynamic geometric latent factors and the dynamic deformable fields as the feature representation, respectively.

DAGSVM-GT [58]	UPSM [59]	HiNet [60]	Multi-stream CNN [61]	RADAP [62]	MoCoGAN [57]	Our's Dglf	Our's Ddf
82.3	87.7	87.2	85.4	85.6	83.1	85.5	**92.0**

TABLE 4
Network architectures of deformable generators for images. The kernel size of the deconvolution layer is listed in $[\cdot]$ and the number of output channel is listed in $(\cdot)$. The stride of all the deconvolution layer is 2. The balance parameter α is set to 0.625

Layer	Appearance Generator Network	Geometric Generator Network
1	$Z^a \sim \mathcal{N}(0, I_{64})$	$Z^g \sim \mathcal{N}(0, I_{64})$
2	FC, $(4 \times 4 \times 128 \times \alpha)$	FC, $(4 \times 4 \times 128)$
3	Deconv+ReLU, $[3 \times 3]$, $(64 \times \alpha)$	Deconv+ReLU, $[3 \times 3]$, (64)
4	Deconv+ReLU, $[3 \times 3]$, $(32 \times \alpha)$	Deconv+ReLU, $[3 \times 3]$, (32)
5	Deconv+ReLU, $[5 \times 5]$, $(16 \times \alpha)$	Deconv+ReLU, $[5 \times 5]$, (16)
6	Deconv+Tanh, $[5 \times 5]$, (3)	Deconv+Tanh, $[5 \times 5]$, (2)

ometric 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 propose two schemes. In the first scheme, 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. In the experiments of deformable generator for images, we adopt this scheme, and set α to 0.625 in our experiments. The appearance and geometric generator's structures are shown in Table 4.

In the second scheme, we explicitly scale the weights between the two generators with a balance parameter β at runtime to control the tradeoff. Specifically, we set $\hat{w}_i^a = w_i^a / c$ and $\hat{w}_i^g = w_i^g / c \times \beta$, where w_i^a and w_i^g are the weights for the appearance and geometric generators, c is the per-layer normalization constant from He's initializer [63]. This dynamical updating scheme for scaling the two set of weights can control the tradeoff between the two generators more flexible, because, in scheme 1, the number of filter is an integer, which results in the limited choice of the balance parameter α, while, in scheme 2, the values of the weights are decimals which results in the infinite number of the choice for the balance parameter β. In the experiments of the dynamic deformable generator for the video sequences, we adopt the second scheme, and set β to 0.3 in our experiments. The corresponding appearance and geometric generator's structures are shown in Table 5.

TABLE 5
Network architectures of dynamic deformable generators for video sequences. The kernel size of the deconvolution layer is listed in $[\cdot]$ and the number of output channel is listed in $(\cdot)$. The stride of the appearance generator network's first 4 deconvolution layer is 2, for the last deconvolution layer is 1 . The stride of the geometric generator network's first 2 deconvolution layer is 4, for the last deconvolution layer is 1. The balance parameter β is set to 0.3

Layer	Appearance Generator Network	Geometric Generator Network
1	$Z_0^a \sim \mathcal{N}(0, I_{60}), s_0^a = 0$	$Z_0^g \sim \mathcal{N}(0, I_{30}), s_0^g = 0$
2	FC+ReLU, $(4 \times 4 \times 512)$	FC+ReLU, $(4 \times 4 \times 256)$
3	Deconv+ReLU, $[3 \times 3]$, (512)	Deconv+ReLU, $[4 \times 4]$, (128)
4	Deconv+ReLU, $[3 \times 3]$, (256)	Deconv+ReLU, $[4 \times 4]$, (64)
5	Deconv+ReLU, $[3 \times 3]$, (128)	Deconv+Tanh, $[5 \times 5]$, (2)
6	Deconv+ReLU, $[3 \times 3]$, (64)	
7	Deconv+Tanh, $[5 \times 5]$, (3)	
Layer	Appearance Transition Network	Geometric Transition Network
1	$\xi_t^a \sim \mathcal{N}(0, I_3), s_0^a = 0$	$\xi_t^g \sim \mathcal{N}(0, I_{10}), s_0^g = 0$
2	FC+ReLU, (20)	FC+ReLU, (20)
3	FC+ReLU, (20)	FC+ReLU, (20)
4	FC+tanh, (60)	FC+tanh, (30)

6 Conclusion and Future Work

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. We also introduce a dynamic deformable generator model for the spatial-temporal process which disentangle the appearance and geometric information of a video sequence into two groups of independent latent vectors Z_t^a and Z_t^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. The geometric generator can also be generalized to incorporate 3D information of rigid or non-rigid 3D objects.

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 at disentanglement in general.

In our work, we mainly focus on disentangling the overall appearance and geometric information into two (groups of) independent vectors. We may further introduce other regularization terms, such as the mutual information or the total correlation, into the existing training objective to enforce and encourage the latent factors within the appearance group or geometry group to be more interpretable and independent.

In our work, the warping function based on coordinate displacements is hand designed. A refinement is that, besides the warping function, another network representing the residual information can be learned from the data. 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.

In the current work, we focus on extracting the appearance and geometric knowledge from the image or video data. For more general situation, we may further extract the topology information, and thus we will disentangle the appearance, geometry and topology information all together. The topology information may take care of specific structure information. Specifically, different categories of images usually take on different structures, so we can employ the one-hot discrete latent variables to model this kind of topology (or structure) information. From the perspective of unsupervised clustering, by disentangling the topology information, we can cluster the data with similar topology information. We have implement this idea, and our preliminary results on the MNIST dataset suggest that we can learn meaningful deformable generator. Specifically, the topological latent factors can specify the desired categories of digits with similar structure information, while within each topological class, the geometric latent factors can control the shape and view information of the generative images.

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.

References

[1] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[2] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun, "Disentangling factors of variation in deep representation using adversarial training," in *Advances in Neural Information Processing Systems*, 2016, pp. 5040–5048.

[3] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in beta-vae," *arXiv preprint arXiv:1804.03599*, 2018.

[4] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," in *Proc. International Conference on Machine Learning (ICML), Sydney*, 2017.

[5] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman, "Building machines that learn and think like people," *Behavioral and Brain Sciences*, vol. 40, 2017.

[6] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *arXiv preprint arXiv:1612.00410*, 2016.

[7] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv:1809.11096*, 2018.

[8] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *arXiv:1812.04948*, 2018.

[9] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet, "Are gans created equal? a large-scale study," *arXiv:1711.10337*, 2017.

[10] A. Brock, J. Donahue, and K. Simonyan, "Large scale gan training for high fidelity natural image synthesis," *arXiv:1809.11096*, 2018.

[11] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 35–51.

[12] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 172–189.

[13] T. Han, Y. Lu, J. Wu, X. Xing, and Y. N. Wu, "Learning generator networks for dynamic patterns," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, pp. 809–818.

[14] J. Xie, R. Gao, Z. Zheng, S.-C. Zhu, and Y. N. Wu, "Motion-based generator model: Unsupervised disentanglement of appearance, trackable and intrackable motions in dynamic patterns," *arXiv preprint arXiv:1911.11294*, 2019.

[15] F. Locatello, G. Abbati, T. Rainforth, S. Bauer, B. Schölkopf, and O. Bachem, "On the fairness of disentangled representations," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 14 584–14 597.

[16] E. Creager and D. Madras, "Flexibly fair representation learning by disentanglement," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. Long Beach, California, USA: PMLR, 09–15 Jun 2019, pp. 1436–1445.

[17] R. Gómez-Bombarelli, J. N. Wei, and D. Duvenaud, "Automatic chemical design using a data-driven continuous representation of molecules," *ACS central science*, vol. 4, no. 2, pp. 268–276, 2018.

[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[19] E. L. Denton, S. Chintala, R. Fergus *et al.*, "Deep generative image models using a laplacian pyramid of adversarial networks," in *NIPS*, 2015, pp. 1486–1494.

[20] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *ICLR*, 2016.

[21] Z. Li, Y. Tang, and Y. He, "Unsupervised disentangled representation learning with analogical relations," *arXiv preprint arXiv:1804.09502*, 2018.

[22] L. Tran, X. Yin, and X. Liu, "Disentangled representation learning gan for pose-invariant face recognition," in *CVPR*, vol. 3, no. 6, 2017, p. 7.

[23] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[24] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *NIPS*, 2014, pp. 1278–1286.

[25] A. Mnih and K. Gregor, "Neural variational inference and learning in belief networks," in *ICML*, 2014.

[26] A. Kumar, P. Sattigeri, and A. Balakrishnan, "Variational inference of disentangled latent concepts from unlabeled observations," *arXiv preprint arXiv:1711.00848*, 2017.

[27] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2016, pp. 2172–2180.

[28] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[29] Z. Lin, K. K. Thekumparampil, G. Fanti, and S. Oh, "Infogan-cr: Disentangling generative adversarial networks with contrastive regularizers," *arXiv preprint arXiv:1906.06034*, 2019.

[30] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in neural information processing systems*, 2015, pp. 2539–2547.

[31] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud, "Isolating sources of disentanglement in variational autoencoders," in *Advances in Neural Information Processing Systems*, 2018, pp. 2610–2620.

[32] P. W. Hallinan, G. Gordon, A. L. Yuille, P. Giblin, and D. Mumford, *Two-and three-dimensional patterns of the face*. AK Peters/CRC Press, 1999.

[33] T. F. Cootes, G. J. Edwards, and C. J. Taylor, "Active appearance models," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 681–685, 2001.

[34] J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "Fast and exact newton and bidirectional fitting of active appearance models,"

IEEE transactions on image processing, vol. 26, no. 2, pp. 1040–1053, 2017.

[35] J. Kossaifi, L. Tran, Y. Panagakis, and M. Pantic, "Gagan: Geometry-aware generative adverserial networks," *arXiv preprint arXiv:1712.00684*, 2017.

[36] Z. Shu, M. Sahasrabudhe, A. Guler, D. Samaras, N. Paragios, and I. Kokkinos, "Deforming autoencoders: Unsupervised disentangling of shape and appearance," *arXiv preprint arXiv:1806.06503*, 2018.

[37] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[38] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 764–773.

[39] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[40] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European conference on computer vision*. Springer, 2004, pp. 25–36.

[41] D. Sun, S. Roth, and M. J. Black, "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 115–137, 2014.

[42] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.

[43] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2017.

[44] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2017.

[45] T. Han, Y. Lu, S.-C. Zhu, and Y. N. Wu, "Alternating back-propagation for generator network." in *AAAI*, 2017, pp. 1976–1984.

[46] L. Younes, "On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates," *Stochastics: An International Journal of Probability and Stochastic Processes*, vol. 65, no. 3-4, pp. 177–228, 1999.

[47] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, 2003.

[48] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[49] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[50] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image Vision Comput.*, vol. 28, no. 5, pp. 807–813, May 2010.

[51] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. 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*. IEEE, 2010, pp. 94–101.

[52] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[53] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 5, pp. 918–930, 2015.

[54] J. Thewlis, H. Bilen, and A. Vedaldi, "Unsupervised learning of object frames by dense equivariant image labelling," in *Advances in Neural Information Processing Systems*, 2017, pp. 844–855.

[55] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European conference on computer vision*. Springer, 2014, pp. 94–108.

[56] N. Aifanti, C. Papachristou, and A. Delopoulos, "The mug facial expression database," in *11th International Workshop on Image Analysis for Multimedia Interactive Services WIAMIS 10*, April 2010, pp. 1–4.

[57] S. Tulyakov, M.-Y. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1526–1535.

[58] D. Sen, S. Datta, and R. Balasubramanian, "Facial emotion classification using concatenated geometric and textural features," *Multimedia Tools and Applications*, vol. 78, no. 8, pp. 10 287–10 323, 2019.

[59] R. Weber, V. Barrielle, C. Soladie, and R. Seguier, "Unsupervised adaptation of a person-specific manifold of facial expressions," *IEEE Transactions on Affective Computing*, 2018.

[60] M. Verma, S. K. Vipparthi, and G. Singh, "Hinet: Hybrid inherited feature learning network for facial expression recognition," *IEEE Letters of the Computer Society*, vol. 2, no. 4, pp. 36–39, 2019.

[61] J. A. Aghamaleki and V. A. Chenarlogh, "Multi-stream cnn for facial expression recognition in limited training data," *Multimedia Tools and Applications*, pp. 1–22, 2019.

[62] M. Mandal, M. Verma, S. Mathur, S. K. Vipparthi, S. Murala, and D. K. Kumar, "Regional adaptive affinitive patterns (radap) with logical operators for facial expression recognition," *IET Image Processing*, vol. 13, no. 5, pp. 850–861, 2019.

[63] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.