Dual-Generator Face Reenactment

Gee-Sern Hsu  Chun-Hung Tsai  Hung-Yi Wu  
National Taiwan University of Science and Technology, Taipei, Taiwan  
{jison, m10903418, m10703435}@mail.ntust.edu.tw

Abstract

We propose the Dual-Generator (DG) network for large-pose face reenactment. Given a source face and a reference face as inputs, the DG network can generate an output face that has the same pose and expression as the reference face, and has the same identity as the source face. As most approaches do not particularly consider large-pose reenactment, the proposed approach addresses this issue by incorporating a 3D landmark detector into the framework and considering a loss function to capture visible local shape variation across large pose. The DG network consists of two modules, the ID-preserving Shape Generator (IDSG) and the Reenacted Face Generator (RFG). The IDSG encodes the 3D landmarks of the reference face into a reference landmark code, and encodes the source face into a source face code. The reference landmark code and the source face code are concatenated and decoded to a set of target landmarks that exhibits the pose and expression of the reference face and preserves the identity of the source face. The RFG is partially built on the StarGAN2 generator with modifications on the input and layer settings, and with a facial style encoder added in. Given the target landmarks made by the IDSG and the source face as inputs, the RFG generates the target face with the desired identity, pose and expression. We evaluate our approach on the RaFD, MPIE, VoxCeleb1, and VoxCeleb2 benchmarks and compare with state-of-the-art methods.

1. Introduction

Given a source face and a reference face, face reenactment refers to the transformation of the action of the reference face to the source face. The action refers to the pose and facial expression. The challenges are on the similarity between the actions of the reference face and the source face and on the preservation of the source identity after the transformation. It is an active research topic in the fields of computer vision and attracts increasing attention in recent years [23–26]. It has a wide range of applications in the areas such as virtual reality, animation and entertainment.

Various approaches have been proposed in recent years [4, 22–25]. A major family of the approaches is the Landmark-Assisted Generation (LAG), which exploits the facial landmarks to leverage the action transformation and the reenacted face generation [4, 22, 24, 25]. The FReeNet [25] trains a landmark converter to transfer the reference’s landmarks to the source, and trains a generator to make the target face show the reference’s expression, but it cannot handle pose transformation. The FSTH [24] trains an embedder to encoder the source’s landmarks, and a generator to transfer the reference’s action to the source face. More approaches from the LAG family are reviewed in Sec.2.

Different from the existing LAG approaches, our approach explores a dual-generator architecture with one generator to make the ID-preserving 3D landmarks, and the other generator to make the target face satisfy multiple objectives. Due to the embedding of 3D landmarks and the core losses considered in training, our approach can address the large-pose reenactment, which is a challenging problem, but has not received sufficient attention.

There are methods without using landmarks, for example, the MGOS [23] that uses the reconstructed 3D meshes as guidance to learn the optical flow needed for the target face synthesis. Although the progresses made by different approaches are substantial, many issues are yet to be solved. The performance measured by the common metrics, for example the FID, CSIM and SSIM, is still far from ideal. Many approaches have specific issues. For example, the FReeNet only transfers the facial expression, but cannot handle pose transfer. Although the FSTH can transfer both the pose and expression, the facial landmarks used to control the conversion are often inaccurate, damaging the identity preservation. Another important issue is that most approaches only deal with median pose variation (yaw angle $< 45^\circ$) and ignore large/extreme poses.

To address the above issues, we propose the Dual-Generator (DG) network that contains two generators, the ID-preserving Shape Generator (IDSG) and the Reenacted Face Generator (RFG). Given a source face $I_s$ and a reference face $I_r$ as inputs, the IDSG transfers the pose and expression of the reference face $I_r$ to the source face $I_s$ and
generates the target landmark estimate \( \hat{l}_t \). The RFG takes the target landmark estimate \( \hat{l}_t \) and the source \( I_s \) as inputs, and generates the reenacted face \( \hat{I}_t \) that shows the same action as of the reference face \( I_r \), but has the same identity as of the source \( I_s \). To handle large-pose references, we embed a 3D-landmark detector and consider an objective function to capture the pose-dependent local shape variation from frontal to profile. We train the DG network on the dataset with full pose variation so that the landmark motion and identity preservation across large pose can be learned.

We summarize the contributions of this work as follows:

- The ID-preserving Shape Generator (IDSG) is verified effective in generating an identity-preserving facial shape with the desired pose and expression.
- The Reenacted Face Generator (RFG) is verified effective in generating an identity-preserving target face with the desired pose.
- Different from most approaches in the LAG family that use 2D landmarks, we embed 3D landmarks with a loss function to capture visible local shape variation so that the large-pose face reenactment can be handled.
- Better performance than state-of-the-art approaches, based on the evaluations on the RaFD, MPIE, VoxCeleb1, VoxCeleb2 benchmarks.

Our code, model and more qualitative results are available on https://github.com/AvLab-CV/Dual_Generator_Face_Reenactment. In the following, we first review the related work in Sec. 2, then the proposed approach in Sec. 3, then the experiments for performance evaluation in Sec. 4, and then a conclusion in Sec. 5.

2. Related Work

Many approaches have been proposed in recent years [4, 18, 22, 23, 25]. A major family of the approaches is the Landmark-Assisted Generation (LAG), which exploits facial landmarks to leverage the expression and pose conversion, followed by the reenacted face generation [22, 24, 25]. There are approaches without using landmarks, for example, the Mesh Guided One-Shot (MGOS) [23] and the X2Face [21]. However, most approaches only concern median pose variation, i.e., the yaw angle \(< 45^\circ\) and ignore large/extreme poses. The proposed approach belongs to the LAG family, but it can handle large/extreme poses, in addition to the common median pose variation.

The ReenactGAN [22] employs an encoder to encode faces into a boundary latent space defined by the heatmaps of facial landmarks. A boundary-based transformer is made to convert the reference’s boundary to the source’s boundary, and an identity-specific decoder synthesizes the transformed boundary to the reenacted face. Although the ReenactGAN can generate good quality target faces, it needs to retrain a new face boundary transformer and decoder when applied to an unseen identity. The Few-Shot Talking Head (FSTH) [24] is made of an embedder network, a generator and a discriminator for activating few-shot learning. The embedder network converts faces into personalized embedding vectors, which are entered into the layers of the generator to make the desired reenacted faces. The FReeNet [25] is made of a landmark converter and a generator for facial expression transfer. The landmark converter transfers the landmark features of the source and reference into the target landmarks with the reference’s expression. The generator takes the transferred target landmarks and the source face for reenactment. The FReeNet only transfers the facial expression and does not transfer the pose, so the reenacted face is in the same pose as of the source, imposing a big limitation on the application. The PuppeteerGAN [4] consists of a sketch network for pose retargeting and a coloring network for appearance transformation. The former takes the source’s segmentation mask and the reference’s landmarks to generate the target segmentation mask and landmarks, which are taken by the latter to make the target preserve the source identity with the reference’s action.

Some approaches do not belong to the LAG family, and consider different annotations or keypoints for capturing the pose and expression transformation. To improve identity preservation, the MGOS [23] uses reconstructed 3D meshes to learn the optical flow needed for the target face synthesis. The learning is based on the optical flow directly from 3D dense meshes, and able to provide the sufficient shape and pose information to reconstruct the source’s expression and pose. The First Order Motion (FOM) model [18] consists of a keypoint detector, a motion network and a generator. The motion network takes the motion representation to generate the dense optical flow from the reference to the source. The generator takes the optical flow and an occlusion map to combine the source appearance and the reference motion to make the desired target face. The X2Face [21] consists of an embedding network and a driving network. The embedding network learns a face representation across source faces with differing poses and expressions and the driving network learns a pixel sampler to convert pixels from the source face to generate the target faces.

As the RFG module in the proposed DG network is developed based on the StarGAN2 [6], we give it a brief review. The StarGAN2 is proposed to address the issues of the StarGAN [5], which learns a deterministic mapping in each visual domain and does not capture the multi-modal nature of the data distribution over multiple domains. The StarGAN2 replaces the domain label in the StarGAN with a domain specific style code to represent the styles of a specific domain. It includes two modules, the mapping network and the style encoder. Both modules have multiple output branches, each of which provides a style code for a specific domain. The StarGAN2 generator learns to synthesize im-
ages over multiple domains by using the style codes.

3. Proposed Approach

The Dual-Generator (DG) network is composed of two primary modules, the ID-preserving Shape Generator (IDSG) and the Reenacted Face Generator (RFG). The configuration is shown in Figure 1. The IDSG consists of a face encoder $E_f$, a facial landmark detector $F_l$, a landmark encoder $E_l$ and a landmark decoder $R_l$. Given a source face $I_s$ and a reference face $I_r$ as inputs, it generates a set of target landmark estimate $\hat{l}_t$ as output. The RFG consists of a face generator $G_f$ and a style encoder $E_s$. Taking the target landmark estimate $\hat{l}_t$ and the source face $I_s$ as inputs, the RFG generates the desired target face $\hat{I}_t$ so that $\hat{l}_t$ and $I_s$ have the same identity, and $\hat{I}_t$ and $I_r$ have the same action in terms of the pose and expression. Both IDSG and RFG are trained for self-reenactment with ground-truth $I_t$ and $\hat{l}_t$ available, then trained for cross-ID reenactment (cross-reenactment) for handling unseen subjects. The details of the above components and modules are given in the following sections. See Supplementary Materials for more details about the network architectures and settings.

3.1. ID-preserving Shape Generator

The IDSG (ID-preserving Shape Generator) is designed to transform the pose and expression of the reference face $I_r$ to the source face $I_s$ in terms of facial landmarks. The problem is formulated as the transformation of the reference facial landmark $l_r$ to the target landmark estimate $\hat{l}_t$ so that $\hat{l}_t$ preserves the identity characteristics of the source $I_s$ but exhibits the pose and expression of the reference $I_r$.

To solve this problem, we design an encoder-decoder landmark generator $G_l = [E_l, R_l]$, where $E_l$ denotes the landmark encoder and $R_l$ is the landmark decoder. At the training phase, the landmark generator $G_l$ works with a landmark discriminator $D_l$ and a landmark-based subject classifier $C_l$. The discriminator $D_l$ verifies the quality of the landmarks made by $G_l$ by distinguishing the generated landmarks from the actual landmarks obtained on the source images. The subject classifier $C_l$ classifies the landmarks of the reference faces according to the subjects in the reference dataset, i.e., $C_l$ classifies the individuals by considering their corresponding landmarks.

Except for the above four major components, $E_l$, $R_l$, $D_l$ and $C_l$, the IDSG incorporates a 3D facial landmark detector $F_l$ and a face encoder $E_f$. Both networks are off-the-shelf models and not updated during training. We use the FAN (Face Alignment Network) [1] as the 3D landmark detector $F_l$, and the feature embedding layers of the VGFace2 [3] as the face encoder $E_f$. The landmark detector $F_l$ detects the 3D landmarks of a 2D face, and labels each landmark as visible or invisible across pose, allowing us to develop the visible local shape loss for handling large-pose reenactment. The face encoder $E_f$ provides the identity loss required to optimize the landmarks generated by the IDSG. The details of the major modules are given below.

- $E_l$ is an MLP made of five fully connected (fc) layers and a leaky ReLU activation function applied to each fc layer, and it generates an action code $\gamma$ to represent the pose and expression of a set of landmarks.
- $R_l$ is an MLP made of five fully connected (fc) layers with five fc layers and leaky ReLU activations. It takes the action code $\gamma$ concatenated with the facial ID code $r_s$ to generate the estimated target landmark $\hat{l}_t$.
- Both the landmark discriminator $D_l$ and subject classifier $C_l$ are structured in the same way as the landmark encoder $E_l$ with the same dimension at input (due to the same dimension of the landmark input) but different dimension at output. The output dimension of $D_l$ is one, for distinguishing the generated landmarks from the real ones; while the output dimension of $C_l$ is the number of subjects to identify in the training set.

We do not just generate the target landmark estimate $\hat{l}_t$, but also the reference landmark estimate $\hat{l}_r$ by entering the reference faces as the source faces during training. One of the novelties in this study is about the loss functions, especially the visible local shape loss, which enables the shape switch across large pose. We consider the following losses: the adversarial loss, the visible local shape loss, the action loss, the subject class loss and the localization loss.

Adversarial Loss To make the target landmark estimate $\hat{l}_t = G_l (I_s, I_r)$ exhibit an actual set of landmarks, the fol-
lowing adversarial losses are needed for training the landmark generator \(G_t\) and discriminator \(D_l\):

\[
L_{G_t}^{adv} = -\mathbb{E}_{l_r \sim p(l_r), l_t \sim p(l_t)} \log [1 - D_l (G_t(l_t, I_s))] \quad (1)
\]

\[
L_{D_l}^{adv} = \mathbb{E}_{l_r \sim p(l_r)} \log [D_l (l_r)] + \mathbb{E}_{l_t \sim p(l_t), l_r \sim p(l_r)} \log [1 - D_l (G_t(l_t, I_s))] \quad (2)
\]

**Visible Local Shape Loss** The visible local shape (VLS) loss \(L_{vls}\) is proposed for two objectives. One is to capture the shape variation of the target landmark estimate \(\hat{l}_t\) across large pose, e.g., one eye occluded when rotating the face to \(> 45^\circ\) in yaw, and reappearing when rotating back. The other one is to make \(\hat{l}_t\) far apart from the reference landmark \(l_r\), while making the estimated reference landmark \(\hat{l}_r\) closer to the real reference landmark \(l_r\) simultaneously, as \(\hat{l}_t\) is made for the source \(I_s\) which must be made further away from the reference \(l_r\), while \(\hat{l}_r\) is made for the reference \(I_r\) which must be made closer to \(l_r\).

We divide the landmarks into five groups for five local regions, namely the left eye, right eye, nose, mouth and face contour. As the landmark coordinates given by the 3D landmark detector \(F_t\) can be used to label visible and invisible landmarks, we can learn the variation of the visible/invisible landmarks across large pose and minimize the following VLS loss for each region during training.

\[
L_{vls}^k = \left\| I_{vls}^k - \tilde{I}_{vls}^k \right\|_1 + \sigma_k \quad (3)
\]

where \(L_{vls}^k\) is the VLS loss defined for Region-\(k\), \(k = 1, 2, ..., 5\) for left eye, right eye, nose, mouth and face contour, respectively; \(v = 0, 1\) is the visibility indicator; \(\sigma_k\) is a margin parameter determined in the experiment. We only compute \(L_{vls}^k\) for the visible landmarks, i.e., \(v = 1\). As the landmark detector \(F_t\) can number each landmark in a specific order regardless of the pose, we group the landmarks for each region by their numbers.

The generation of \(\hat{l}_t\) is driven by the concatenated \([\gamma, r_s]\), and the generation of \(\hat{l}_r\) driven by the concatenated \([\gamma, r_r]\), where \(r_s\) and \(r_r\) are the facial ID codes of the source \(I_s\) and reference \(I_r\), respectively. The VLS loss \(L_{vls}\) constrains the generation of \(l_r\) and \(\hat{l}_r\) by using the reference landmark code \(\gamma\), awards the closeness between \(l_r\) and \(\hat{l}_r\), and penalizes the similarity between \(l_r\) and \(\hat{l}_t\).

**Action Loss** To better duplicate the pose and expression of the reference, we minimize the following action loss \(L_a\) that computes the difference between the landmark codes of the reference landmark and target landmark estimate.

\[
L_a = \left\| E_t(l_t) - E_t(l_r) \right\|_1 \quad (4)
\]

**Subject Class Loss** We use the subject classifier \(C_l\) to compute the following subject class loss \(L_{C_l}\) to make \(l_t\) preserve the subject identity in the shape space.

\[
L_{C_l} = \mathbb{E}_{l_t \sim p(l_t)} [-\log P(s_i | C_l(l_t))] \quad (5)
\]

where \(s_i\) is the ID label of the reference face \(I_r\).

**Localization Loss** To make the generated landmarks located at the desired locations, the localization loss \(L_l\) is exploited to minimize the distance between \(l_t\) and \(l_r\) and the distance between \(l_t\) and \(\hat{l}_t\) when the ground truth \(l_t\) can be available at the self-reenactment training phase.

\[
L_l = \left\| \hat{l}_t - l_t \right\|_1 + \left\| \hat{l}_t - l_r \right\|_1 \quad (6)
\]

Note the differences between the global additive setup in (6) and the local adversarial setup in (3), and the different desired objectives.

The following weighted sum of the above five losses is minimized when training the IDSG.

\[
L_{IDSG} = L_{G_t}^{adv} + \lambda_a L_{vls} + \lambda_a L_a + \lambda_c L_{C_l} + \lambda_l L_l \quad (7)
\]

where \(\lambda_a, \lambda_c, \lambda_a\), \(\lambda_a\) are the weights to be determined in the experiments.

### 3.2. Reenacted Face Generator

The Reenacted Face Generator (RFG) takes the target landmark estimate \(\hat{l}_t\) and the source image \(I_s\) as input, and generates the reenacted face \(\hat{I}_t\) as output. The desired \(\hat{I}_t\) must be of the same identity as of the source face \(I_s\), and in the same pose and expression as of the reference face \(I_r\). It is composed of an encoder-decoder generator \(G_f\) and a style encoder \(E_s\). During training, \(G_f\) and \(E_s\) learn along with a face discriminator \(D_f\) and a shape discriminator \(D_s\) to produce the desired target face \(\hat{I}_t\). The details of the above modules are presented below.

- The style encoder \(E_s\) consists of six downsampling residual blocks and aims to extract the facial style code \(s_s = E_s(I_s)\) from the source \(I_s\), \(s_s\) will be entered to the layers of the generator \(G_f\) to preserve the source identity at the generated target \(\hat{I}_t\).
- The generator \(G_f\) consists of four downsampling residual blocks, four intermediate residual blocks and four upsampling residual blocks. The AdaIN [11,12] is applied to enter the facial style code \(s_s\) into the last two intermediate residual blocks and all upsampling residual blocks to make the target face \(\hat{l}_t = G_f(\hat{l}_t, s_s)\). We enter \(\hat{l}_t\) into \(G_f\) in the form of a landmark map, which is a binary image of the landmarks with each neighboring landmark pair connected by an edge.
- The face discriminator \(D_f\) and shape discriminator \(D_s\) have the same structure as of the style encoder \(E_s\), but with both 1D output for discriminating the generated from the real. The input to \(D_f\) is \(\hat{I}_t\), and the input to \(D_s\) is \(F_t(\hat{I}_t)\).
Although the generator \( G_f \) is built on the StarGAN2, the differences include the layer settings for entering the style signal \( s_s \), the source format in a binary map, the discriminator settings and the loss functions. We consider the following loss functions when training the RFG for self-reenactment with the ground-truth target \( I_t \) is available.

**Adversarial Loss** Force the generated target \( \hat{I}_t \) to comply with two requirements: 1) \( \hat{I}_t \) must appear as a real face with the same identity as of the source face \( I_s \); 2) \( \hat{I}_t \) must be in the same action as of the reference \( I_r \). The following adversarial losses for \( G, D_f \) and \( D_s \) are needed to meet these requirements:

\[
\mathcal{L}^{adv}_G = - \mathbb{E}_{l_t \sim p(l_t), I_s \sim p(I_s)} \log \left[ 1 - D_f \left( G(\hat{I}_t, I_s) \right) \right] \tag{8}
\]

\[
\mathcal{L}^{adv}_D_f = \mathbb{E}_{l_t \sim p(l_t)} \log \left[ D_f \left( I_t \right) \right] + \mathbb{E}_{l_s \sim p(l_s), I_s \sim p(I_s)} \log \left[ 1 - D_f \left( G(\hat{I}_t, I_s) \right) \right] \tag{9}
\]

\[
\mathcal{L}^{adv}_D_s = \mathbb{E}_{F_t(l_t) \sim p(F_t(l_t))} \log [D_s \left( F_t(I_t) \right)] + \mathbb{E}_{F_t(l_t) \sim p(F_t(l_t))} \log \left[ 1 - D_s \left( F_t(\hat{I}_t) \right) \right] \tag{10}
\]

**Attribute Loss** To make the image attributes of the generated target \( \hat{I}_t \) close to those of the ground-truth target \( I_t \), we exploit the following pixel-wise \( L_1 \) loss \( \mathcal{L}_{at} \):

\[
\mathcal{L}_{at} = \left\| \hat{I}_t - I_t \right\|_1 \tag{11}
\]

**Identity Loss** To preserve the source identity of \( I_s \) at the generated face \( \hat{I}_t \), we use the face encoder \( E_f \) formed by the feature embedding layers of the VGGFace2 [3] to compute the following identity (ID) loss via cosine similarity.

\[
\mathcal{L}_{id} = 1 - \cos(E_f(\hat{I}_t), E_f(I_s)) \tag{12}
\]

**Style Consistency Loss** To make the style encoder \( E_s \) generate the same facial style code \( s_s \) to the source \( I_s \) and the generated target \( \hat{I}_t \), we exploit the following loss:

\[
\mathcal{L}_{st} = \left\| E_s(\hat{I}_t) - E_s(I_s) \right\|_1 \tag{13}
\]

**Landmark Loss** To make the generated target face \( \hat{I}_t \) appear in the desired action, we exploit the following landmark loss \( \mathcal{L}_{lm} \) to minimize the distance between \( \hat{l}_t \) and the landmarks detected on \( I_t \).

\[
\mathcal{L}_{lm} = \left\| F_t(\hat{I}_t) - \hat{l}_t \right\|_1 \tag{14}
\]

The full objective function for training the RFG is a weighted sum of the above loss functions:

\[
\mathcal{L}_{RFG} = \mathcal{L}^{adv}_G + \lambda_{at} \mathcal{L}_{at} + \lambda_{id} \mathcal{L}_{id} + \lambda_{st} \mathcal{L}_{st} + \lambda_{lm} \mathcal{L}_{lm} \tag{15}
\]

where \( \lambda_{at}, \lambda_{id}, \lambda_{st}, \lambda_{lm} \) are the weights to be determined.

4. **Experiment**

We first introduce the datasets, then the evaluation and implementation details, and then an ablation study on different settings of the DG network. A comparison with state-of-the-art approaches is presented with the performance on both the normal and large-pose settings.

4.1. **Datasets and Implementation Details**

We consider both the constrained and unconstrained datasets. The RaFD [14] and MPIE [9] are the constrained datasets that offer ground truth for target poses and expressions; the VoxCeleb1 [16] and VoxCeleb2 [7] are the unconstrained (aka in-the-wild) datasets.

**RaFD** The Radboud Faces Database (RaFD) [14] consists of 8,040 pictures collected from 67 subjects. Each subject has 8 expressions in 3 gaze directions and 5 different poses. All images were resized to 256² pixels, and we used the FAN to detect the 68 3D landmarks on each face. We followed the same settings as in the FReeNet [25]. The training set was formed by 67 subjects with 8 facial expressions in 3 gaze directions and 5 different poses. For performance evaluation, we synthesized 100 reenacted images for each source identity with 100 reference images randomly selected from other identities, resulting in 6,700 reenacted images for the 67 subjects.

**MPIE** The MPIE offers more than 750k images for 337 subjects in 15 poses, 6 expressions and 20 lighting conditions. It is selected for the evaluation on large pose reenactment. We followed the same setup as that in [2]. The training set is formed by 200 subjects with all poses and 5 lighting condition and 4 expressions, and the rest 137 subjects form the testing set. The training set is used for self-reenactment, and the testing set is used for cross-reenactment. We design two test protocols for cross-reenactment. One synthesized 100 reenacted images for each source identity in the testing set with 100 reference images randomly selected from other identities. The other repeated the experiments but each source face with yaw < 30° and reference faces with yaw > 60°. The latter is called MPIE (Large Pose) in the experiments.

**VoxCeleb1** The VoxCeleb1 dataset [16] contains over 100k utterances for 1,251 celebrities, extracted from the YouTube videos, and is divided into the training and testing sets. In our experiment, all images were extracted from the videos sampled at 1 fps, resized to 256² pixels, and each with 3D landmarks detected by the FAN. We followed the experimental protocol reported in the FSTH [24], and trained all models on the training set. For the performance evaluation, we fine tuned all models by using 8 frames randomly selected from the 50 videos in the test set, and tested on the 32 hold-out frames of the same 50 videos (fine-tuning and the hold-out frames do not overlap).
VoxCeleb2 The VoxCeleb2 [7] is an extension of the VoxCeleb1. It contains over 1 million utterances for 6,112 celebrities, and is divided into the training and testing sets. We extracted images from the videos at 25 fps and processed the images in the same way as performed for the VoxCeleb1. We again followed the protocol reported in the FSTH [24] for the experiments.

Evaluation Metrics Multiple metrics are selected to test the photo-realistic quality and identity preservation of the generated images, including the Frechet-Inception Distance (FID) [10], the Structured Similarity (SSIM) [20] and Cosine Similarity (CSIM). The FID evaluates the photorealistic quality by measuring the distribution distance between the features extracted from the real and generated images. The feature is extracted by using the last average pooling layer of the Inception-V3 [19]. The SSIM measures the low-level similarity of the generated images to the ground-truth images. The CSIM measures the identity preservation in the generated images by using the similarity between the facial features extracted from the source and generated images. We use the feature embedding layers of the ArcFace [8] to extract the facial features, and compute the cosine similarity.

Implementation Details We trained the IDSG and RFG, independently; and merged them for testing. We began with self-reenactment with minimum two images per identity for training, and one image used as source and the other as reference. Based on the model trained for self-reenactment, we retrained it for cross-reenactment with references replaced by other identities.

We trained the IDSG module from scratch with the objective defined in (7). The following parameters were determined from a comparison study. The margins \([m_1]_{1,\ldots,5}\) for the VLS loss in (3) were selected as 0.05, 0.05, 0.1, 0.05 and 0.2, respectively. The weights in (7) were settled as \(\lambda_{l} = 0.5, \lambda_{c} = 1, \lambda_{sl} = 10, \lambda_{a} = 1\). We also trained the RFG module from scratch with the objective given in (15). To compute the identity loss in (12), we extracted the 2048D feature from the last fully connected layer of the VGGFace2 built on the ResNet50 [3]. The weights in (15) were selected as \(\lambda_{at} = 10, \lambda_{id} = 10, \lambda_{st} = 1\) and \(\lambda_{tm} = 1\). Our programs were written in the Pytorch deep learning framework [17]. All experiments were run with batch size 4 on a Ubuntu 18.04 with NVIDIA RTX Titan GPU. We used the Adam [13] optimizer with \(\beta_1 = 0.01, \beta_2 = 0.99\). The learning rates for the two modules were \(1e^{-5}\) and \(1e^{-4}\), respectively.

4.2. Ablation Study

To better determine the settings of the loss functions for the IDSG and the RFG, we selected the RaFD as the dataset to determine the settings for the loss functions, and the MPIE (Large Pose) for demonstrating the effect of the IDSG. Both the RaFD and MPIE (Large Pose) offer different faces of the same pose and expression so that the ground truth for the target action can be available for comparison.

Loss Functions for the IDSG We computed the Average Coordinate-wise Error (ACE) of the landmarks generated by the IDSG with different loss settings. We define a baseline that only considers the adversarial loss \(L_{adv}\) and classification loss \(L_{cls}\), and other loss functions are cumulatively added to the baseline. The performance comparison in ACE is given in Table 1. The ACE decreases when the localization loss \(L_{l}\) and the action loss \(L_{a}\) are added to the baseline. When the VLS loss \(L_{cls}\) is added on, the ACE is substantially improved. Due to page limit, please see Supplementary Materials for qualitative comparisons.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>SSIM↑</th>
<th>FID↓</th>
<th>CSIM↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL: (L_{adv}^D + L_{at})</td>
<td>0.503</td>
<td>58.61</td>
<td>0.211</td>
</tr>
<tr>
<td>+ (L_{sd})</td>
<td>0.643</td>
<td>12.01</td>
<td>0.775</td>
</tr>
<tr>
<td>+ (L_{st})</td>
<td>0.662</td>
<td>9.92</td>
<td>0.803</td>
</tr>
<tr>
<td>+ (L_{tm})</td>
<td>0.707</td>
<td>5.59</td>
<td>0.844</td>
</tr>
<tr>
<td>DG (+ (L_{adv}^s))</td>
<td>0.726</td>
<td>3.99</td>
<td>0.862</td>
</tr>
</tbody>
</table>

| Table 1. Average Coordinate-wise Error (ACE) on RaFD dataset for different loss settings on the IDSG. Baseline (BL) refers to the model with adversarial loss \(L_{adv}^D\) and classification loss \(L_{cls}\) only. |

Loss Functions for the RFG Table 2 shows the FID, SSIM and CSIM when each loss function is cumulatively added to the RFG baseline, which only considers the face discriminator \(D_f\) and the attribute loss \(L_{at}\). The identity loss \(L_{id}\) can significantly improve the image quality and identity preservation. The style consistency loss \(L_{st}\) and landmark loss \(L_{tm}\) also enhances the overall quality and performance. The additional shape discriminator \(D_s\) further improves the generated quality and identity preservation, as demonstrated by all three metrics, especially the FID. See Supplementary Materials for qualitative comparisons.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>SSIM↑</th>
<th>FID↓</th>
<th>CSIM↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>BL: (L_{adv}^D + L_{at})</td>
<td>0.503</td>
<td>58.61</td>
<td>0.211</td>
</tr>
<tr>
<td>+ (L_{sd})</td>
<td>0.643</td>
<td>12.01</td>
<td>0.775</td>
</tr>
<tr>
<td>+ (L_{st})</td>
<td>0.662</td>
<td>9.92</td>
<td>0.803</td>
</tr>
<tr>
<td>+ (L_{tm})</td>
<td>0.707</td>
<td>5.59</td>
<td>0.844</td>
</tr>
<tr>
<td>DG (+ (L_{adv}^s))</td>
<td>0.726</td>
<td>3.99</td>
<td>0.862</td>
</tr>
</tbody>
</table>

| Table 2. RFG performance for different losses cumulatively added on to the baseline (BL) with \(L_{adv}^D + L_{at}\) on the RFG. Top four rows with \(D_f\) only, last row with \(D_s\) added on. |

Influence of IDSG Figure 2 shows the effect of the IDSG sampled from the experiment on MPIE (Large Pose). When the poses of the source and reference are close to frontal, the RFG alone performs well in identity preservation with the source \(I_s\) and the reference landmark \(L_r\) as input, i.e., the shape information is all given by the reference without using the IDSG. But the facial contour generated looks similar to the reference, instead of the source. This can be a serious issue when the reference is in large pose. As the cases shown in Figure 2, the RFG mistakes the reference’s landmark as a mouth-open pattern and makes the reenacted faces open mouth. When using the target land-
Figure 2. The second row shows the reenacted faces made by the RFG with reference landmark $l_r$, i.e., without using the IDSG; the third row made by the DG ($=\text{IDSG+RFG}$).

Figure 3. Comparison with several SOTA approaches for self-reenactment.

Table 3. Comparison of self-reenactment performance with state-of-the-art methods on the VoxCeleb1 dataset

<table>
<thead>
<tr>
<th>Method (N)</th>
<th>SSIM↑</th>
<th>FID↓</th>
<th>CSIM↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>X2Face [21]</td>
<td>0.75</td>
<td>56.5</td>
<td>0.18</td>
</tr>
<tr>
<td>FSTH [24]</td>
<td>0.74</td>
<td>29.5</td>
<td>0.19</td>
</tr>
<tr>
<td>FOM [18]</td>
<td>0.723</td>
<td>25.0</td>
<td>0.813</td>
</tr>
<tr>
<td>PuppeteerGAN [4]</td>
<td>0.725</td>
<td>33.6</td>
<td>0.717</td>
</tr>
<tr>
<td>MGOS [23]</td>
<td>0.739</td>
<td>n.a.</td>
<td>0.822</td>
</tr>
<tr>
<td>DG</td>
<td>0.761</td>
<td>22.1</td>
<td>0.831</td>
</tr>
</tbody>
</table>

Table 4. Cross-reenactment performance compared with SOTA methods on VoxCeleb2, RaFD, MPIE and MPIE (Large Pose)

<table>
<thead>
<tr>
<th>Method (N)</th>
<th>SSIM↑</th>
<th>FID↓</th>
<th>CSIM↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>VoxCeleb2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOM [18]</td>
<td>0.53</td>
<td>54.78</td>
<td>0.714</td>
</tr>
<tr>
<td>DG</td>
<td>0.54</td>
<td>51.79</td>
<td>0.721</td>
</tr>
<tr>
<td>RaFD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOM [18]</td>
<td>0.723</td>
<td>9.37</td>
<td>0.801</td>
</tr>
<tr>
<td>DG</td>
<td>0.726</td>
<td>4.79</td>
<td>0.862</td>
</tr>
<tr>
<td>MPIE</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOM [18]</td>
<td>0.58</td>
<td>28.34</td>
<td>0.714</td>
</tr>
<tr>
<td>DG</td>
<td>0.65</td>
<td>16.55</td>
<td>0.780</td>
</tr>
<tr>
<td>MPIE (Large Pose)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FOM [18]</td>
<td>0.38</td>
<td>62.88</td>
<td>0.382</td>
</tr>
<tr>
<td>DG</td>
<td>0.61</td>
<td>25.66</td>
<td>0.711</td>
</tr>
</tbody>
</table>

mark estimate $\hat{l}_r$, i.e., the reference’s landmark rectified by the IDSG, the performance is considerably improved.

4.3. Comparison with State-of-the-Art Methods

The DG network with the best settings confirmed in the ablation study is compared with state-of-the-art approaches for handling both the self-reenactment and cross-reenactment. We ran the same experiments for the approaches with code available. For the approaches without code, we duplicate the results and image samples in their papers for comparison.

**Self-Reenactment** Table 3 shows the self-reenactment performance on the VoxCeleb1 dataset compared with the X2face [21], FSTH [24], FOM [18], PuppeteerGAN [4] and MGOS [23]. The DG net achieves the best scores in all three metrics. Figure 3 shows the qualitative comparison with some of the approaches and the ground truth. The DG demonstrates better performance in identity preservation and facial expression similarity to the ground truth. However, as those samples are all close to frontal pose, the performance for reenactment across large pose needs a different evaluation. Although the X2face, FOM and FSTH have released models/codes, only the FOM model offers similar results as reported in the paper according to our tests. We are unable to duplicate the performance of the X2face and FSTH as similar to what they reported in their papers by using their models/codes. The samples in Figure 3 are photo-copied from their papers.

**Cross-Identity** Table 4 shows the cross-reenactment performance on the VoxCeleb2, RaFD, MPIE datasets. As mentioned in Sec. 4.1, the MPIE has two testing protocols and one is for testing large-pose performance. Very few methods report performance for cross-reenactment, and we only found that the FReeNet [25] presents the performance on the RaFD. The performance of the FOM in Table 4 is based on the model released by the authors which we have retrained on MPIE and MPIE (Large Pose). The DG net claims the best performance in all three metrics on all benchmarks, including the MPIE Large-Pose. Figure 4 shows a qualitative comparison with the faces made by the FReeNet and FOM. Note the FReeNet can only handle facial expression transfer but cannot deal with pose transfer, as the generated faces are all in the same pose as of the source. The FOM can deliver good results to the sources in frontal pose, but does not work for the sources with large poses. Please see Supplementary Materials for more qualitative comparisons for cross-reenactment performance.
Figure 4. Cross-reenactment comparison with FReeNet and FOM on the RaFD. Top row shows the references. Those enclosed by red bounding boxes are made by the DG net.

Figure 5. Cross-reenactment samples on the MPIE dataset.

4.4. Performance for Large-Pose Reenactment

Figure 5 shows cross-reenactment samples on the MPIE, compared with the ground truth. To demonstrate the performance for handling large-pose reenactment, the references are selected for large pose differences from the source face and a few references are in extreme poses. The reenacted faces well preserve the source identity and exhibit the poses and expressions of the references. For comparison purpose, we trained the DG network on the combination of the MPIE and VoxCeleb1 training sets, and tested the cross-reenactment performance on the test sets. Figure 6 shows several cases with the source faces from the VoxCeleb1 and the extreme-pose reference from MPIE. The comparison includes results made by the FOM, as it shows satisfying performance for sources in frontal pose. However, the FOM is unable to handle the source with extreme pose. The DG network performs well for the source with extreme pose if it is trained on the MPIE, which offers sufficient data in large/extreme poses for learning. The performance deteriorates if the training set does not contain MPIE, which offers a sufficient amount of large-pose training data.

5. Conclusion

We propose the Dual-Generator (DG) network for face reenactment. It is composed of two generators, one for generating an identity-preserving facial shape with the reference’s pose and facial expression, and the other for generating the desired reenacted face. As most approaches do not particularly consider large-pose reenactment, the proposed DG network address this issue by incorporating a 3D landmark detector into the framework and considering a loss function to capture visible local shape variation across large pose. Experiments verify that the DG network outperforms state-of-the-art approaches in the action ranges considered by most existing approaches, and perform satisfactorily for large-pose reenactment.
References

[1] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In ICCV, 2017. 3