Supplementary Material
Make a Face: Towards Arbitrary High Fidelity Face Manipulation

Abstract

This documents provides additional information regarding our main paper, network architecture and detailed implementations . Furthermore, we provide extend results to illustrate the advantages of our method due to space limits. The supplementary material is organized as follows:

- *Sec. 1 illustrates the basic formulation and usage of C-VAE on solving our problems.*
- *Sec. 2 presents effects of weight normalization in detail, including loss curves and further analysis.*
- *Sec. 3 describes the network architecture used in experiments.*
- *Sec. 4 provides more experiments details, including detailed datasets descriptions, AMT perceptual scores used and specific implementations.*
- *Sec. 5 shows additional experiments results on more datasets, including more general non-human cases. We also demonstrate more applications using our model, including face animation and video generation.*

1. Modeling faces based on latent shape and appearance

Given samples x from a face dataset, it is straightforward to use VAEs which aim at modeling the data likelihood $p(x)$. As VAEs assume that the data points x around a low-dimensional manifold could be parameterized by latent embeddings z. The decoder is employed to obtain x based on latent representation z. As a result, in order to make posterior $p(z|x)$ tractably computable, a encoder is applied to approximate it with a distribution $q(z|x)$. Thus, VAEs are based on the formulation:

$$\log p(x) - D_{KL}[q(z|x), p(z|x)] = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{KL}[q(z|x), p(z)] \quad (1)$$

However, for tasks like face manipulation or conditional image synthesis, synthesized images are conditioned by two factors y, z. Let y denotes head pose information, including motion, expression, yaw angles and other spatial information. z should refer to internal appearance information. Apparently, according to the formulation in equation 1, z couldn't be well separated and thus are not able to satisfy our goal in dealing with tasks like face manipulation.

As discussed in the main paper, in order to disentangle the two factors, y and z and learn the distribution $p(x|y,z)$. As y could provide additional spatial information(landmarks in our case), the latent variable z could be inferred by maximizing their conditional log likelihood:

$$logp(x|y) = log \int_z p(x,z|y)dz \geq \mathbb{E}_q \frac{p(x,z|y)}{q(z|x,y)} \quad (2)$$

$$= \mathbb{E}_q log \frac{p(z|,y,z)p(z|y)}{q(z|x,y)} \quad (3)$$

According to equation 2, the evidence lower bound (ELBO) now depends on the conditional prior $p(z|y)$. Thus more semantic correlations between spatial and appearance information could be captured.

2. Weight Normalization and its Effects

As mentioned before, for model training, weight normalization [10] is utilized. As has been tailored in GAN-based approaches, different normalization tricks are used mostly in order to stabilize adversarial training steps.

For VAE-based generative models, which are particularly noise-sensitive, potential noises brought by normalization tricks [4] would reduce the diversity of generated samples significantly, while the absence of normalization tricks would affect model's convergence. Hence we tailor this strategy to both *encoder* and *decoder* of our model. As can be seen from Fig. 1, models trained with weight normalization have faster convergence and lower reconstruction loss than the one without WN over training iterations. We also compare using 'vanilla' weight normalization with using weight normalization and sub pixel(pixel shuffle) convolution as ablative study. The blue curve represents using WN and PS is covered by the red one. It shows that

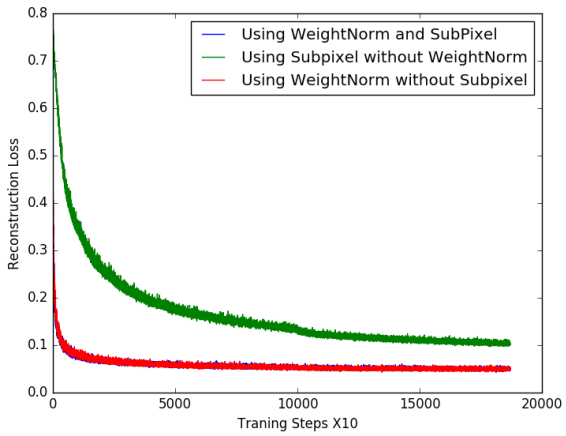

Figure 1: Training Curve w/wo pixel-shuffle and WN.

weight normalization help training converge and enhance model's generative power. These curves also supports that pixel shuffle and weight normalization helps synthesis respectively in a different way as we conveyed in the main paper.

Specifically, Batch normalization re-calibrates the mean and variance of intermediate features to solve the problem of internal covariate shift during deep nets training, its formulation doesn't fit generative tasks very well with two drawbacks: on the one hand, for high-fidelity/resolution image synthesis, small mini-batch sizes are always applied due to limited GPU memory. On the other hand, especially for VAE-based 'noise-sensitive' variants, with imposed distribution constrains and noise gained by normalization tricks statistics, VAE's performance would be constrained significantly.

Formulation Assume the output $\mathbf{y}$ is with the form:

$$\mathbf{y} = \mathbf{w} \cdot \mathbf{x} + b, \tag{4}$$

where $\mathbf{w}$ is a k-dimensional vector representing weight, b is the bias term, $\mathbf{x}$ is a k-dimensional input features. WN re-parameterizes the weight using

$$\mathbf{w} = \frac{g}{||\mathbf{v}||}\mathbf{v}, \tag{5}$$

where v is a k-dimensional vector, g is a scalar, and $||\mathbf{v}||$ denotes the Euclidean norm of $\mathbf{v}$. With this formalization, we will have $||\mathbf{w}|| = g$, independent of parameters $\mathbf{v}$.

In practice, normalization tricks could stabilize GAN's training and aid model's performance. While for VAE training, which optimize the KL divergene where the loss quantities could be large through training, weight normalization is applied to avoid potentialities of collapse. Intuitively, WN is just a normalization trick and doesn't effect given representation power. This phenomenon has also been observed in image SR tasks [13] recently. Compared to BN, Weight Normalization addresses these drawbacks using BN, also eases the difficulty of training VAEs.

3. Network Architecture

In this section, we provide details regarding the network architectures in our networks. Our network contains two branches of encoder and one decoder, which is illustrated in Fig. 2. For all the experiments which synthesize the size of 256×256 images, we use 6 residual blocks for down-sample. Detailed visualization can be found at Fig. 2 and specific choices of parameters can be found at Tables. 1 2.

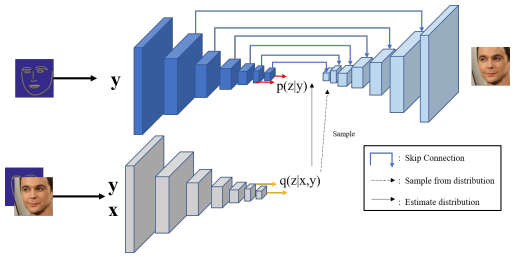

Figure 2: Our basic model architecture. Skip connection on the structure branch is incorporated. As mentioned, we down-sample 6 times for a image during our most experiments. During training, we sample the appearance distribution.

Layer	Kernel Size	Output Channel	Output Size
Input	-	3	256
Convolution	3	64	256
Pooling	2	64	128
Convolution	3	128	128
Pooling	2	128	64
Convolution	3	256	64
Pooling	2	256	32
Convolution	3	512	32
Pooling	2	512	16
Convolution	3	512	16
Pooling	2	512	8
Convolution	3	512	8
Pooling	2	512	4
Output	-	512	4
level1 output	-	512	8

Table 1: Details parameters of encoders, upper-branch encoder share the same architecture of the lower one.

Layer	Kernel Size	Output Channel	Output Size
Input	-	512	4
Convolution	3	2048	4
Pixel Shuffle	-	512	8
Concat	-	1024	8
Convolution	3	2048	8
Pixel Shuffle	-	512	16
Concat	-	1024	16
Convolution	3	1024	16
Pixel Shuffle	-	256	32
Concat	-	512	32
Convolution	3	512	32
Pixel Shuffle	-	128	64
Concat	-	256	64
Convolution	3	256	64
Pixel Shuffle	-	64	128
Concat	-	128	128
Convolution	3	12	128
Pixel Shuffle	-	3	256

Table 2: Details of decoders. Note that for fused feature is not included in the parameters since it depends on how many levels of latent features to be fused.

4. Experiments Details

In this section, more details of experiments are provided. We first introduce all datasets we used in details in Sec. 4.1. Then specific settings and usage of evaluation protocols are shown in Sec. 4.2. We also provide more implementation details in Sec. 4.3.

4.1. Dataset descriptions

We evaluate our model on RafD [7], MultiPIE [2], CelebA [8] and 3D synthesized datasets, containing both indoor and in-the-wild scenarios. We also conduct experiments on none-human datasets, including hands [1] and cats datasets, in order to demonstrate the generalizability of our approach. Followings are detailed descriptions of each dataset we used in experiment.

For real-word human faces datasets: (1)**RafD** [7] dataset consists of $4,824$ images collected from 67 participants with 8 facial expressions in three different gaze directions for each person. (2) **MultiPIE** [2] consists of 20 illumination conditions, 13 poses within 90 yaw angles and 6 expressions of 337 subjects. These two datasets are captured in controlled environments. (3) **CelebA** [8] is a large-scale face attributes dataset in uncontrolled environment, which contains $202,599$ face images of celebrities with large pose variations and background clutter.

For **3D synthetic** face data, we employ 3DMM [9] to create 17920 distinct faces image with 10 shapes indicate 10 different types of facial structures. Each face has been rendered at 8 types of random texture and 4 different lighting environment have been used. Also, 8 expressions are chose for each faces. The orientation of faces is allowed to vary in azimuth from $0°$ to $180°$ by increments of $30°$.

For non-human datasets. We use **hand** and **cat**. (1) **Hands**. The landmarks of each hands is obtained using pre-trained hand detector. Then the skeleton is interpolated using the landmarks obtained. (2)**Cats**, we crawled cats image from the Internet. Then, each cat face is annotated with 18 landmarks.

For each datasets, 90% identities are used for training. and the left 10% are fed into the model for testing.

4.2. Evaluation Metrics

We evaluate the realism perceptual quality and diversity of synthesis results using different metrics. Human subjective study on Amazon Mechanical Turk (AMT) are conducted for realism measurement. Detailed illustrations of metrics we used are provided:

TS (TrueSkill) [3] and **FR** (Fool Rate) are reported. Specifically, **TS** is a skill-based ranking system containing multiple players that allows us to compare our method with other methods. In practice, we randomly select two images generated by three algorithms. The participant is asked to decide which image demonstrates better quality. **FR** is used to evaluate the fidelity of generated images based on direct comparison with real images. Given two samples, a participant is asked to distinguish the real one. Note that since pix2pixHD [12] solely depends on condition and are not able to preserve the reference person. Thus we didn't conduct Fool Rate evaluation using pix2pixHD.

4.3. Implementation Details

Before training, all faces are cropped and aligned to 256×256. For the network structure, details can be found at Fig. 2. PReLU [11] is used as the activation function. Each BN module in residual block is replaced by WN module and sub-pixel convolution is used for up-sampling.

At training, we used Adam [6] optimizer with learning rate of 0.001, beta1 0.5, beta2 0.999 and batch size 4 on a single GTX TITANX GPU. For the feature matching loss using VGG-19, we simply follow perceptual loss [5]. The coefficient of KL divergence loss is set to 1.

5. Additional Experiments Results

Followings are additional experiments for comprehensive demonstration.

5.1. Comparison with Pix2pixHD

we also conduct comparison with pix2pixHD [12], a fully-supervised and general image-to-image translation framework which can also work well in our settings. The result can be found at Fig. 3. As shown, pix2pixHD are likely to generate images with certain blur mainly due to its poor generalizability to unseen inputs. Besides, as it only

Figure 3: Comparison with Pix2PixHD [12] on RaFD. The left image denotes the source image. The first row shows 8 target boundary references. The second row shows images synthesized by pix2pixHD. The third row presents image generated by our algorithm. pix2pixHD takes boundary reference as input and thus are not able to keep source identity.

takes boundary reference as its input, it is not able to keep the identity of source image, making generated results 'uncontrollable'. In contrast, our approach are able to synthesis photo-realistic faces with appealing perceptual quality as well as maintaining the source appearance.

Besides, our model is also able to generate diverse outputs given fixed boundary reference input. As shown in Fig. 4, we sample appearance and condition on boundary maps. Compared to Pix2PixHD's 'same' outputs when tested multiple times, our model has much higher diversity.

5.2. Face Manipulation on 3D Synthetic Data

In order to better observe variations on appearance during manipulation, also expect light/texture-preserving nature of our model, we also conduct qualitative experiments on 3D synthetic face manipulation in Fig. 6.

As shown, factors including skin,texture,lighting can be well preserved, which better testify to the great disentanglement of our model.

5.3. Face Manipulation from single image

In this subsection, we conduct face manipulation from a single image. By randomly modify the landmarks/boundary maps, our model are able to generate diverse and photo-realistic results which maintains the given appearance. Manipulating results can be found at Fig. 7.

5.4. Non-human Cases

Experiments on non-human datasets are aimed at testifying the generalizability of our model. We conduct experiments on hands, cats dataset with the same hyper-parameter settings used in face datasets. Besides, we also test our CelebA-trained model on other stylized faces.

We test our model which trained using real world human face dataset(CelebA) on unreal stylized faces(e.g. cartoon, sketch) in Fig. 5, presenting the robustness of our model.

5.5. Face Animation and Video Generation

We also conduct experiments on face animation and video generation. ***Our method can perform video face animation from a single image***. Given a single source image and an target video with its landmarks(could be manipulated), by feeding those original landmarks and source image to our model frame by frame, the generated videos could be animated face results where the source face acts like the the face in the original video. Our framework can generate high-quality video as well as maintaining correlations on **Facial Action Consistency**, as shown in Fig. 8.

5.6. Missing Identity Issues in Arbitrary Face Manipulation

As discussed in main paper, arbitrary face manipulation potentially leads to identity changing issues. In this section, we provide a quantitative analysis about identity in Table. 3, using Verification accuracy to compare with input's neutral face. During experiment, The model is trained on CelebA dataset and tested on Rafd dataset. Each input face image is assigned with 8 target boundary maps. We report the accuracy of both using same source landmarks and random selecting from the dataset. It illustrate that landmark information implicit covers partial identity information. Our model is capable of preserving identities when the landmarks belongs to same source. Note that there exists no prior knowledge about identity, e.g. pre-trained face recognition model, indicating that our framework could implicitly encode identity information beyond appearance.

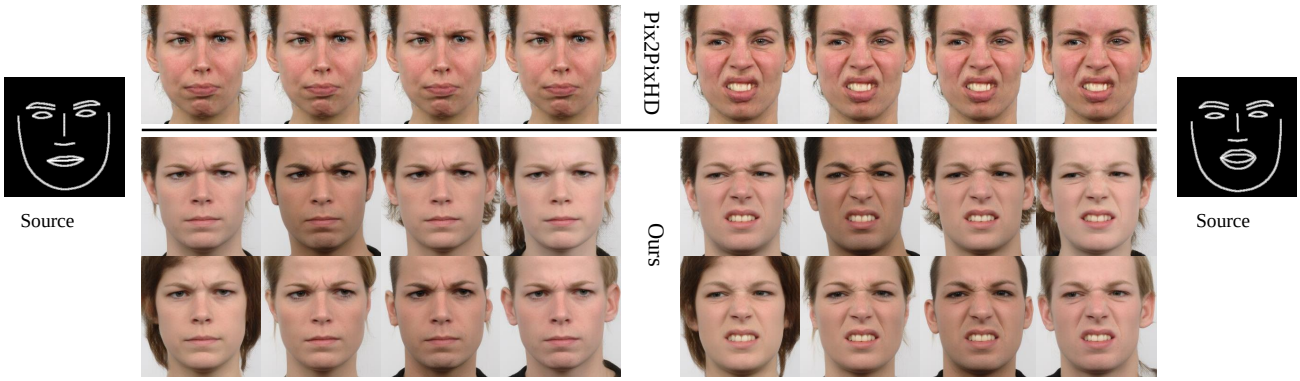

Figure 4: Comparison with pix2pixHD [12] on the diversity of sampling results. Two source boundary maps are taken as input. The first row shows images synthesized by pix2pixHD, each source genrating 4 samples. The second and third rows present images generated by our model with randomly sampled 8 appearances from the dataset each source.

Input($\times 8$)	Frontal	Profile	
		$\pm 45°$	$\pm 90°$
Random Source	29.3%	20.7%	10.1%
Same Source	95.8%	87.3%	61.2%

Table 3: Face verification results on Rafd dataset. Generator is trained on CelebA. Input column denotes the landmark source from the original person or random.

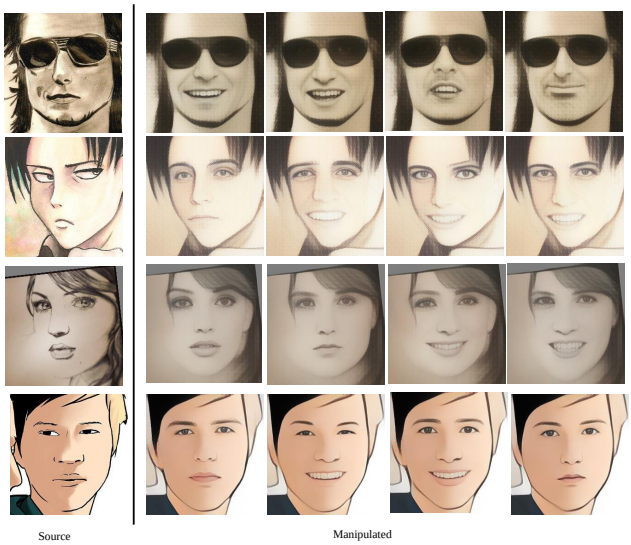

Figure 5: Testing results on special stylized face(cartoon, sketch). Note that the model is trained on real world CelebA dataset and haven't seen peculiar faces before. The first line on the left presents four source faces. For each image given, four manipulated results are provided to the right.

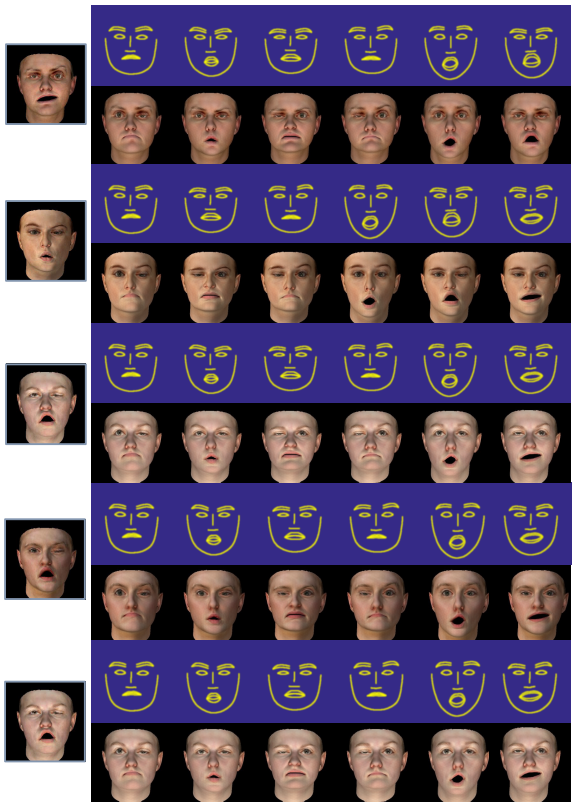

Figure 6: Face manipulation results on 3D image by modify the location of its landmarks. Images on the left with blue boxes are source images. Images on the right each odd row represents target boundary maps and each even row shows corresponding synthesized results.

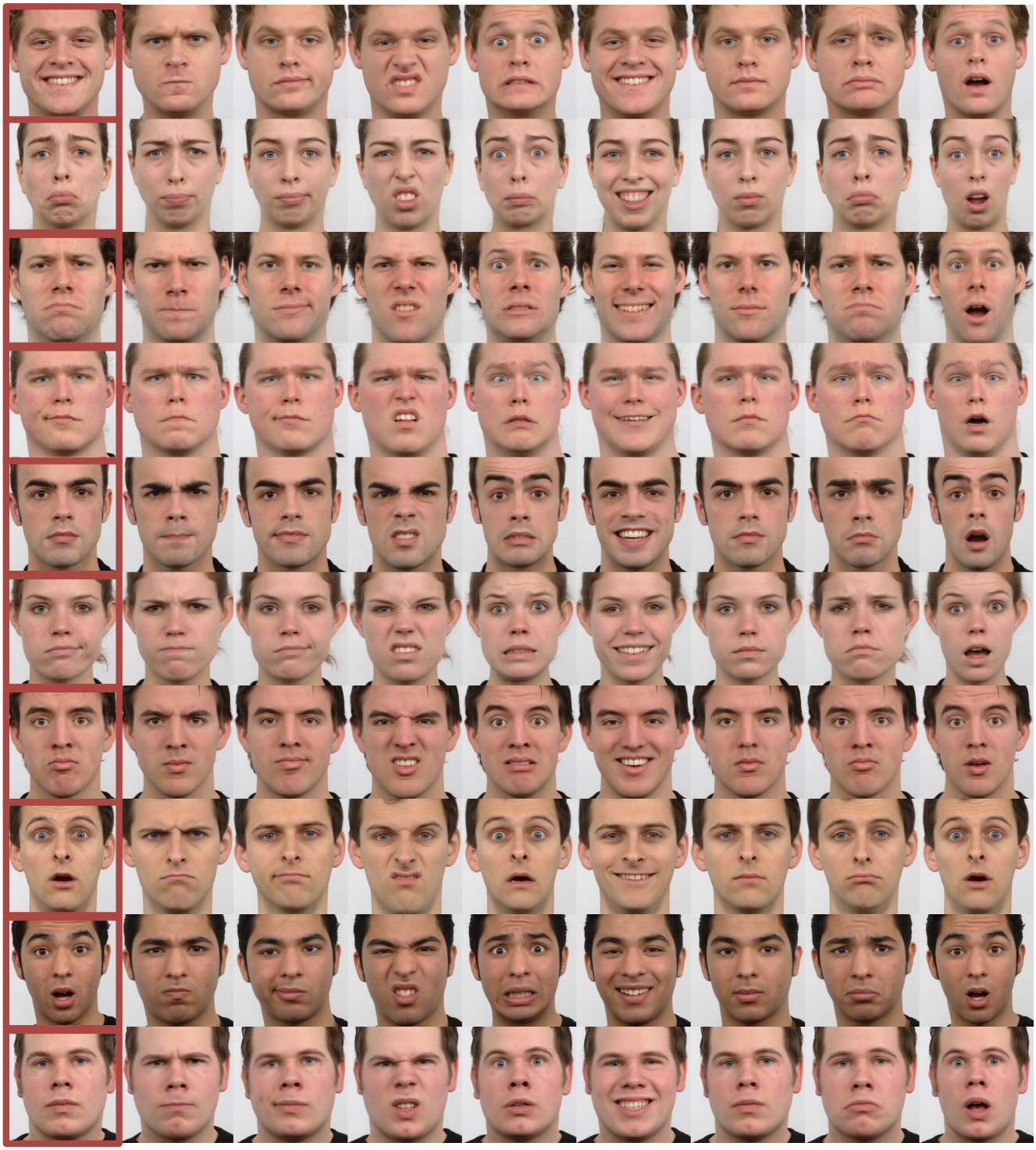

Figure 7: Extensive results about face manipulation on RafD dataset. For each row, the left most one is the input source face. Then the model is fed with the person's boundary maps of 8 target expressions. On the right are 8 synthesized results under target poses. Note that boundary maps each row used are different and unseen during training.

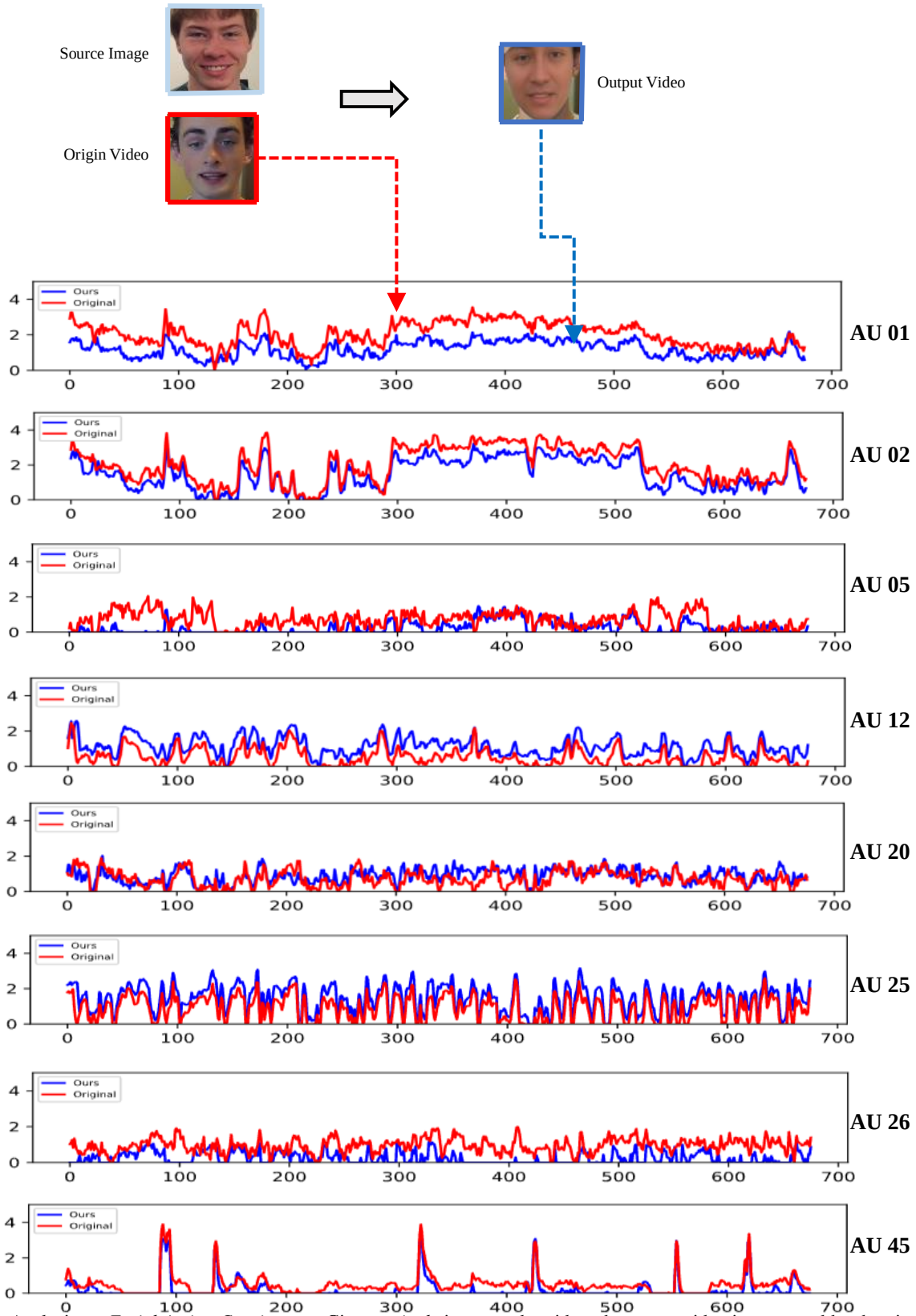

Figure 8: Analysis on *Facial Action Consistency*. Given a *single* image and a video, the output video is generated by the given image and landmarks in the video frame by frame. We use a facial action detector to obtain responses from our model. 8 representative AUs are selected to show the correlation of AUs response between our result and the source video. Each graph shows AUs response vs time.

References

[1] Mahmoud Afifi. Gender recognition and biometric identification using a large dataset of hand images. *arXiv preprint arXiv:1711.04322*, 2017. 3

[2] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 2010. 3

[3] Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill: a bayesian skill rating system. In *NIPS*, 2007. 3

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 1

[5] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 3

[6] Diederick P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[7] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 2010. 3

[8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015. 3

[9] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *AVSS 2009*, 2009. 3

[10] Tim Salimans and Diederik P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NIPS*, 2016. 1

[11] Ludovic Trottier, Philippe Gigu, Brahim Chaib-draa, et al. Parametric exponential linear unit for deep convolutional neural networks. In *ICMLA*, 2017. 3

[12] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 3, 4, 5

[13] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018. 2