Depth-Aware Generative Adversarial Network for Talking Head Video Generation

-Supplementary Material-

Fa-Ting Hong[1] Longhao Zhang[2] Li Shen[2] Dan Xu[1*]

[1]Department of Computer Science and Engineering, HKUST [2]Alibaba Cloud

fhongac@cse.ust.hk, longhao.zlh@alibaba-inc.com, lshen.lsh@gmail.com, danxu@cse.ust.hk

A. Additional Network and Training Details

A.1. Loss details

Perceptual loss $\mathcal{L}_P$. To ensure that the generated images are similar to their corresponding ground truths, we use a multi-scale implementation introduced by FOMM [5]. Specifically, we first downsample the ground truth and the output image to 4 different resolutions (*i.e.* 256×256, 128×128, 64×64 and 32×32). We denote R_1,R_2,R_3,R_4 as the generated images, and G_1,G_2,G_3,G_4 as the corresponding ground truths of the four different resolutions, respectively. Then a pre-trained VGG network is used to extract features from both these downsampled ground truths and the output images. We compute the $\mathcal{L}_1$ distance between the ground truth and output image in different resolutions:

$$\mathcal{L}_P = \sum_{i=1}^{4} \mathcal{L}_1(G_i, R_i) \tag{1}$$

GAN loss $\mathcal{L}_G$. Given the ground truths and the generated images in 256×256 resolution, we adopt an adversarial learning objective function consisting of a least square loss and a feature matching loss introduced in the pix2pixHD [6] to train our DaGAN. Single-scale discriminators are used for training 256×256 images.

Equivariance loss $\mathcal{L}_E$. This loss is utilized to ensure the consistency of the estimated keypoints, which is also adopted by FOMM [5]. Given an image $\mathbf{I}$ and one of its detected keypoint $\mathbf{x}_k$, we perform a known spatial transformation $\mathbf{T}$ on image $\mathbf{I}$, resulting in a transformed image $\mathbf{I_T}$. Therefore, the detected keypoints $\mathbf{x}_{\mathbf{T}(k)}$ on this transformed image $\mathbf{I_T}$ should be transformed in the same way. Thus, for the K detected keypoints from image $\mathbf{I}$, we have:

$$\mathcal{L}_E = \sum_{i=1}^{K} ||\mathbf{x}_k - \mathbf{T}^{-1}(\mathbf{x}_{\mathbf{T}(k)})||_1 \tag{2}$$

*Corresponding author

Keypoints distance loss $\mathcal{L}_D$. To make the detected facial keypoints much less crowded around a small neighbourhood, we employ a keypoints distance loss to penalize the model if the distance between two corresponding keypoints falls below a pre-defined threshold. For every two keypoints $\mathbf{x}_i$ and $\mathbf{x}_j$ in an image, we thus have:

$$\mathcal{L}_D = \sum_{i=1}^{K} \sum_{j=1}^{K} (1 - \mathbf{sign}(||\mathbf{x}_i - \mathbf{x}_j||_1 - \alpha)), i \neq j, \tag{3}$$

where $\mathbf{sign}(\cdot)$ is a sign function, and the α is the threshold of distance. It is set to 0.2 in our work, which shows good performance in our practice.

A.2. Network architecture details of DaGAN

The implementation details of the sub-networks in our model are shown in Fig. 1 and described below.

Face depth network $\mathcal{F}_d$. Our face depth network consists of an encoder and a decoder. The encoder is a ResNet18 network [2] without the final fully connected and pooling layers. The structure of the decoder is illustrated in Fig. 1a, which predicts a depth map with a size of $1 \times 256 \times 256$.

Keypoint estimator $\mathcal{F}_{kp}$. In the training process, we concatenate the RGB image and its corresponding depth map to form an RGB-D input with a size of $4 \times 256 \times 256$, while the ouputs are K keypoints $\{\mathbf{x}_{\tau,n}\}_{n=1}^{K}, \mathbf{x}_{\tau,n} \in \mathbb{R}^{1 \times 2}$. The detailed structure of the keypoint estimator is shown in Fig. 1b.

Occlusion estimator $\mathcal{T}$. We utilize the occlusion estimator to predict an occlusion map to filter out the regions that should be inpainted, and a motion flow mask for weighting the motion field. As illustrated in Fig. 1c, there are two heads at the end to predict these two parts.

Feature encoder $\mathcal{E}_I$. In Fig. 1f, to preserve low-level texture of the image, we only apply two DownBlocks to construct the feature encoder $\mathcal{E}_I$ in the feature warping module.

Depth encoder $\mathcal{E}_d$. The architecture of our depth encoder $\mathcal{E}_d$ in the cross-modal attention module is shown in Fig. 1g.

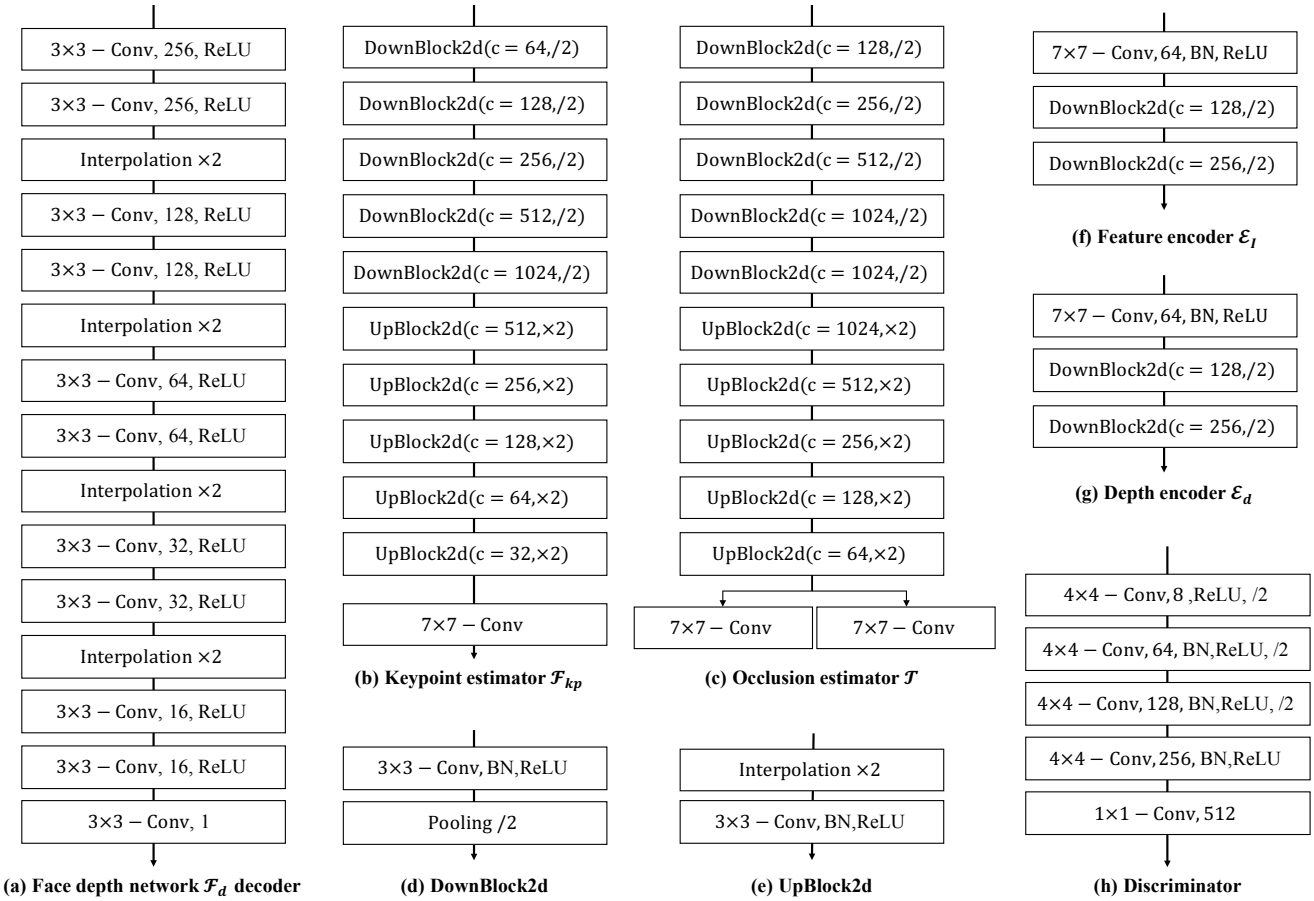

Figure 1. Architecture details of each components in our model. The "DownBlock2d" (Fig. 1d) contains a convolutional layer with 3×3 kernel, a batch normalization layer, a ReLU activation layer, and an average pooling layer that downsamples the input. The interpolation layer in "UpBlock2d" (Fig. 1d) is utilized to upsample the image. The symbol "/2" in other sub-networks indicates an average pooling layer to downsample the input.

The structure is the same as $\mathcal{E}_I$, and thus we can make the features learned from both modalities with the same level of representation power.

Discriminator $\mathcal{D}$. The architecture of our discriminator (Fig. 1h) is inspired by FOMM [5]. The input image is first down-sampled four times, and then passed through a convolutional layer with a kernel size of 1×1, and we finally output a prediction map with a size of $512 \times 26 \times 26$. Moreover, we collect the intermediate feature maps and feed them into the GAN loss $\mathcal{L}_G$.

B. Additional Experiment Details

B.1. Dataset Details

- **VoxCeleb1** dataset contains videos of 1,251 different identities with a resolution of 256×256. We extract frames for each video and utilized the test split of VoxCeleb1 for evaluating self-reenactment. Following [1, 9], we created the test set by sampling 2,083 image sets from randomly selected 100 videos of the VoxCeleb1 test split.
- **CelebV** dataset contains videos of five different celebrities with widely varying characteristics, which are utilize to evaluate the performance of the models for reenacting unseen targets, similar to the in-the-wild scenarios. Moreover, we uniformly sampled 2000 image sets from CelebV to perform the experiments.

B.2. Compare methods

- **X2Face** [8]. X2Face utilizes a simple framework to warp the image directly. We obtain its results on VoxCeleb1 from a previous work [1].
- **NeuralHead** [10]. NeuralHead adopts an important component from style transfer [3, 4], *i.e.* AdaIN layers [3]. Since a reference implementation is absent, we directly report the replicated results from [1].
- **MarioNETte** [1]. MarioNETte utilizes three components (*i.e.* image attention block, target feature alignment, and landmark transformer) to address the identity preservation problem. We compare with it based on the results reported in the original paper.

Figure 2. Additional qualitative comparison of different methods.

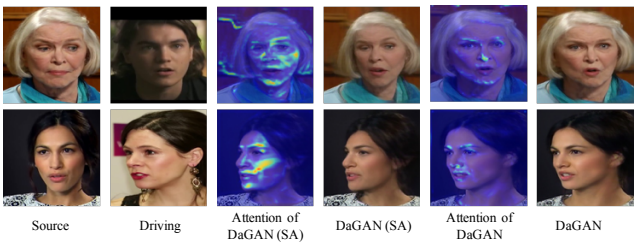

Figure 3. Visualization of attention maps of different methods.

- **FOMM** [5]. FOMM propose a paradigm that aims to detect the keypoints of the face image and model the motion between two images using detected keypoints.
- **MeshG** [9]. MeshG aims to generate a dense face mesh to model a dense motion map using graph convolutional network. As there is no official code available, we only report the its results from the original paper.
- **OSFV** [7]. OSFV provides a novel keypoint generation method. We reimplemented this method according to its published paper and train it on the VoxCeleb1 dataset to compare with the proposed method.

B.3. More results

More explanation of depth-aware attention. Each learned 3D spatial depth point is inherently used as a query for calculating a global self-attention, which is thus depth-aware. Here, we disable the depth in the cross-modal attention module, which then becomes a standard self-attention module, termed as DaGAN (SA). The compared results are shown in Fig. 2 and Table 1. It is clear that, the facial depth is very important for the learning of dense 3D-geometry-guided global self-attention, leading to clearly better generation performance. Additionally, a qualitative comparison in Fig. 3 shows the difference of using and not using depth for the attention learning. Our cross-modal attention can effectively learn to attend to key foreground facial regions (e.g. expression-related keypoint regions), comparing to the one without depth (*i.e.* DaGAN (SA)) which also attends to cluttered backgrounds, further confirming the advantage of dense 3D geometry for overcoming noisy background in generation.

More explanation of our baseline. To better illustrate our baseline method, we select some other samples as shown in Fig. 2, and we compare our baseline with FOMM as shown in Table 1. From the Table 1, the baseline is quantitatively very competitive to FOMM (better than FOMM in CSIM, worse in PRMSE, and comparable on AUCON). More qualitative examples shown in Fig. 2 can further indicate the comparable performance between the baseline and FOMM.

Model	CSIM ↑	PRMSE ↓	AUCON ↑	$\mathcal{L}_1$ ↓	AKD ↓	AED↓
Baseline	**0.688**	5.39	0.657	**0.040**	1.537	0.189
FOMM [5]	0.462	**3.90**	**0.667**	0.043	**1.294**	**0.140**
DaGAN (SA)	0.681	5.18	0.832	0.045	2.015	0.242
DaGAN	**0.723**	**2.33**	**0.873**	**0.036**	**1.279**	**0.117**

Table 1. Results for the baseline and cross-modal self-attention.

More qualitative results. We show more samples in Fig. 4 and Fig. 5. The visualization shows that our DaGAN can produce more natural-looking faces than the other comparison methods. More than that, we also present our generated depth maps of the source images and the driving images. We can observe that our estimated depth maps can effectively distinguish the face foreground area of an image from the background. These robustly predicted depth maps can also verify the effectiveness of our method for self-supervised dense geometry recovery.

References

[1] Sungjoo Ha, Martin Kersner, Beomsu Kim, Seokjun Seo, and Dongyoung Kim. Marionette: Few-shot face reenactment preserving identity of unseen targets. In *AAAI*, 2020. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[3] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 2

[4] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 2

[5] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *NeurIPS*, 2019. 1, 2, 3

[6] 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. 1

[7] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *CVPR*, 2021. 3

[8] Olivia Wiles, A Koepke, and Andrew Zisserman. X2face: A network for controlling face generation using images, audio, and pose codes. In *ECCV*, 2018. 2

[9] Guangming Yao, Yi Yuan, Tianjia Shao, and Kun Zhou. Mesh guided one-shot face reenactment using graph convolutional networks. In *ACM MM*, 2020. 2, 3

[10] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. Few-shot adversarial learning of realistic neural talking head models. In *ICCV*, 2019. 2

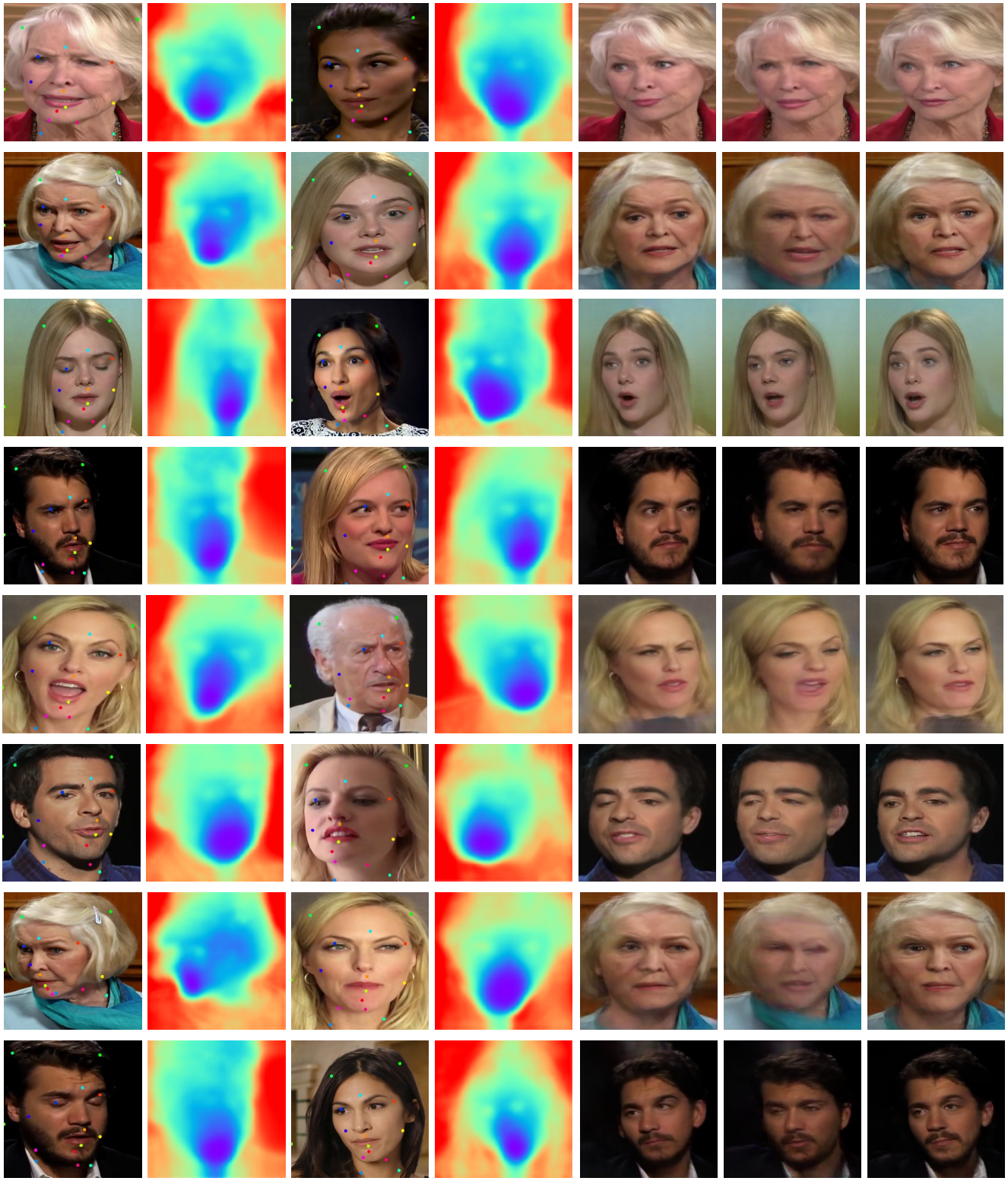

Figure 4. Qualitative comparisons of different methods on cross-identity face reenactment. We also show the predicted face depth maps and detected keypoints of source images and driving images.

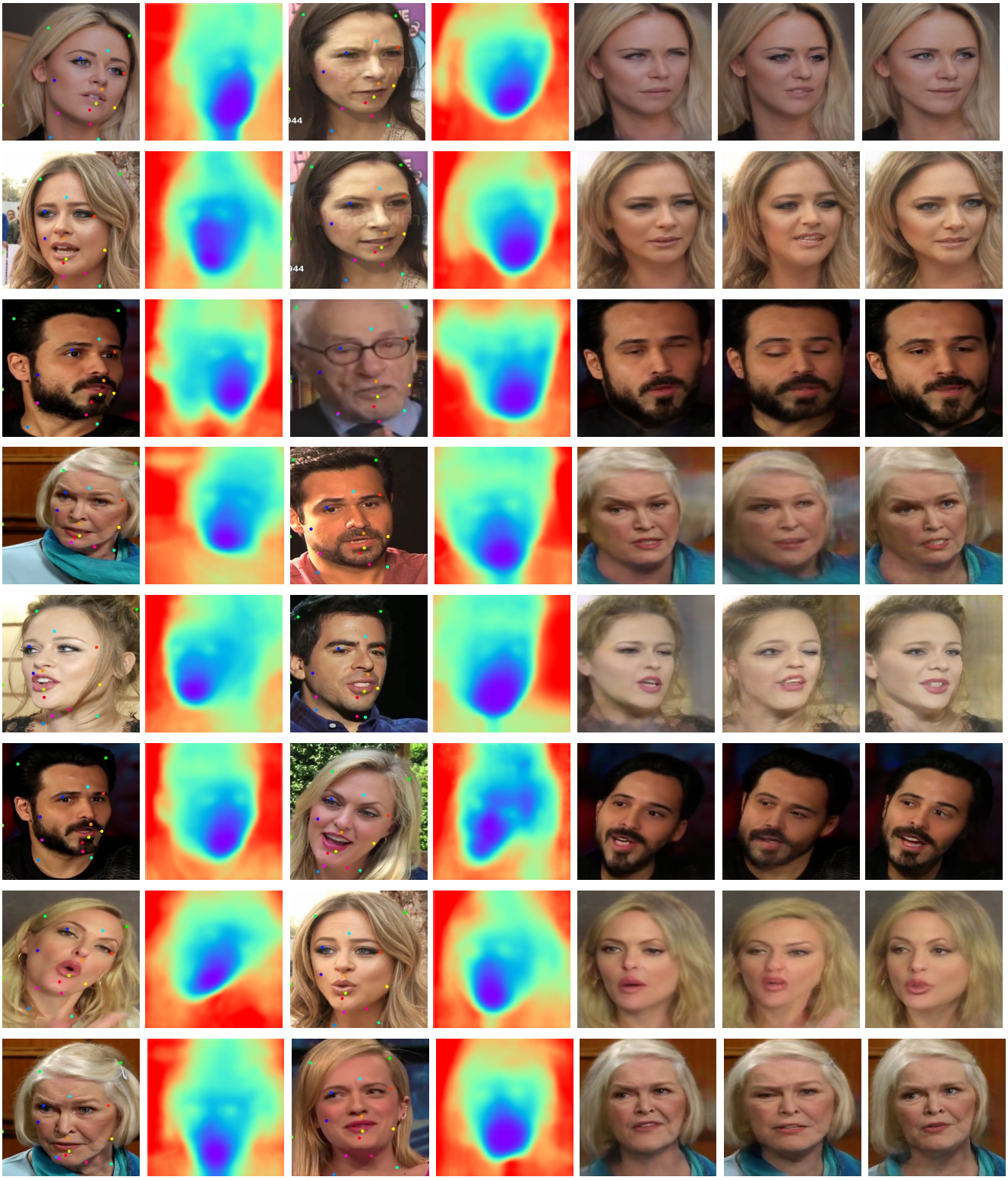

Figure 5. Qualitative comparisons of different methods on cross-identity face reenactment. We also show the predicted face depth maps and detected keypoints of source images and driving images.