High-Fidelity and Freely Controllable Talking Head Video Generation

Yue Gao Yuan Zhou Jinglu Wang Xiao Li Xiang Ming Yan Lu
Microsoft Research
{yuegao, zhouyuan, jinglu.wang, li.xiao, xiangming, yanlu}@microsoft.com

In this supplementary material, we elaborate on the loss functions, implementation details, datasets, additional results on the comparisons with different methods, and more results of free pose and expression editing on wild identities. For the video results, please refer to https://yuegao.me/PECHead.

1. Loss Functions

All the following losses are accumulated on all T frames and normalized by T. For brevity, the subscript t is omitted in the equations.

Pixel-wise Loss $\mathcal{L}_p$. The pixel-wise loss is defined as,

$$\mathcal{L}_p = \mathbb{E}[\parallel \hat{x}^d - x^d \parallel_1], \tag{1}$$

where $\hat{x}^d$ is the synthesized frame and the x^d is the driving frame.

Perceptual Loss $\mathcal{L}_v$. We follow the same setting as depicted in FOMM [21], the perceptual loss is defined as,

$$\mathcal{L}_v = \mathbb{E}[\sum_i \sum_j \parallel \text{VGG}_j(\hat{x}_i^d) - \text{VGG}_j(x_i^d) \parallel_1], \tag{2}$$

where i indicates that the frame is down-sampled by i times, and j is the layer index of the VGG-Net [23].

Learned Landmarks Loss $\mathcal{L}_k$. The learned landmarks loss [26] is used to control the distance of the learned landmarks and prevent the distance falling below a threshold,

$$\mathcal{L}_k = \mathbb{E}[\sum_{z=\{s,d\}} \sum_{i=1}^{K} \sum_{j=1}^{K} (\max(0, \eta - \parallel k_i^z - k_j^z \parallel_2^2)], \tag{3}$$

where η is a preset threshold, we set $\eta = 0.1$.

Equivariance Loss $\mathcal{L}_e$. The equivariance loss [21, 22] is defined as,

$$\mathcal{L}_e = \mathbb{E}[\parallel E(\mathcal{T}(x^s), p^d, e^d) - \mathcal{T}(E(x^s, p^d, e^d)) \parallel_1], \tag{4}$$

where $\mathcal{T}$ is the nonlinear random thin-plate spline (TPS) transformation, x^s is the source frame, p^d and e^d is the head pose and expression of the driving frame x^d.

Warping Loss $\mathcal{L}_w$. The warping loss is designed to make the warped source frame close to the driving frame. It is defined as follows,

$$\begin{aligned}\mathcal{L}_w = \mathbb{E}[0.5\times \parallel \mathcal{W}(x^s, w_{\text{local}}^{s\to d}) - x^d \parallel_1 + \\ 0.5\times \parallel \mathcal{W}(x^s, w_{\text{global}}^{s\to d}) - x^d \parallel_1].\end{aligned} \tag{5}$$

GAN Loss $\mathcal{L}_G, \mathcal{L}_D$. We adopt the Hinge Loss [18] as the adversarial loss [14], and two patch discriminators for different scales are used to achieve better performance [13],

$$\begin{aligned}\mathcal{L}_G &= -\mathbb{E}[D(\hat{x}^d)], \\ \mathcal{L}_D &= \mathbb{E}[\max(0, 1 - D(x^d)) + \max(0, 1 + D(\hat{x}^d))].\end{aligned} \tag{6}$$

2. Implementation Details

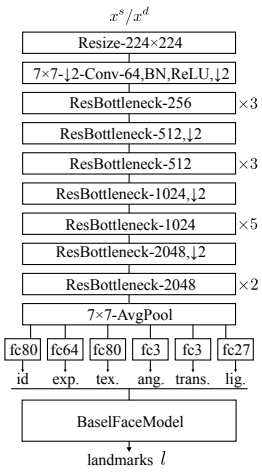
2.1. Model Details

Our method contains four parts, the Face Shape Reconstructor R, the Head Pose-Aware Keypoint Estimator E, the Generator G, and the Multi-Scale Discriminator D. The details of the model structures and sub-modules are shown in Figure 1. The Spectral Normalization (SN) [16] is applied to two discriminators. A Squeeze and Excitation (SE) block [12] is employed in the last layer of E for modeling the correlation among different landmarks. The SE block is also used in the motion estimators. Unlike FOMM [21] and other existing methods [11, 22, 31], we do not estimate the Jacobians. In our experiments, we observe that it is hard to control the head pose with the Jacobians, the same observations are also reported by Wang et al. [26]. The Face Shape Reconstructor R uses ResNet-50 [9] to predict coefficients, and the Basel Face Model (BFM) [1, 8] to further estimate the projected face landmarks.

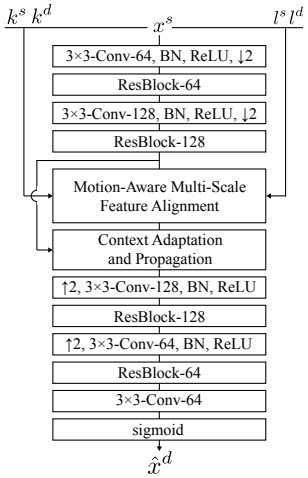
2.2. Datasets Details

Four datasets, *i.e.*, VoxCeleb2 [4], TalkingHead-1KH [26], CelebV-HQ [32], and VFHQ [29], are used in this paper.

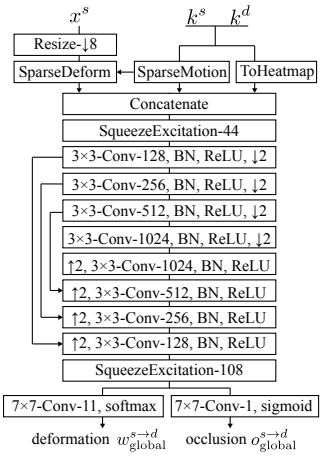
VoxCeleb2. The VoxCeleb2 dataset contains 1M talking-head videos captured from different celebrities. The amount



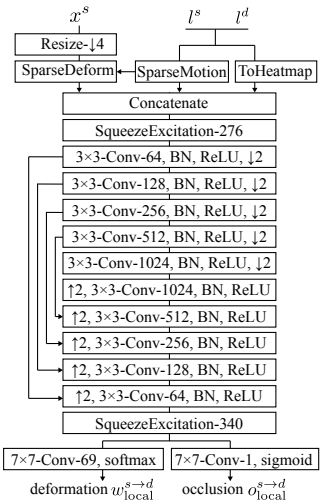
(a). Face Shape Reconstructor R



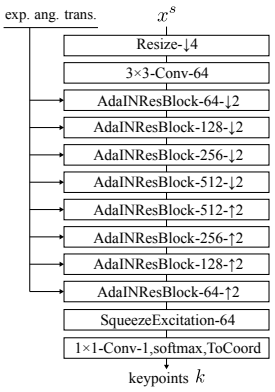
(b). Generator G



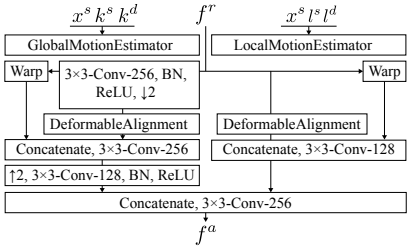
(c). Global Motion Estimator



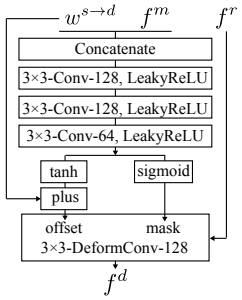
(d). Local Motion Estimator



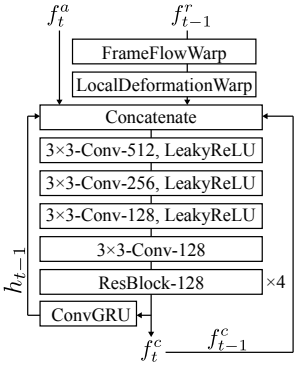
(e). Keypoint Estimator E



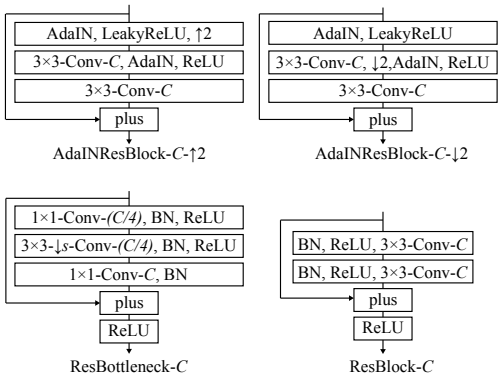
(f). Motion-Aware Multi-Scale Feature Alignment



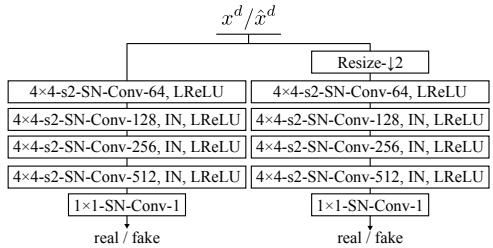
(g). Deformable Alignment



(h). Context Adaptation and Propagation



(i). ResBlock variants



(j). Multi-Scale Discriminator D

Figure 1. The detailed architectures of components in our model.

of videos and identities of this dataset is huge, but the average quality of these videos is limited.

TalkingHead-1KH. The TalkingHead-1KH dataset is composed of about 1,000 hours crowd sourcing videos. Compared with VoxCeleb2, the video frames of TalkingHead-1KH have higher quality and resolution.

CelebV-HQ. The CelebV-HQ provides more than 35K video clips with diverse appearances, actions, and expressions, involving more than 15K identities.

VFHQ. The VFHQ dataset is mainly constructed for video face super-resolution, which contains over 16K high-fidelity clips of diverse interview scenarios, providing the highest frame quality among these datasets.

For TalkingHead-1KH and VFHQ, we follow the approaches used in [26, 29] to split the training and validation sets respectively, and report the performance of our model on the validation sets. For CelebV-HQ, we randomly select 500 videos for validation, as the official validation split is not provided. For VoxCeleb2, we randomly sample 500 videos from the full validation set containing more than 19K videos. The resolution of frames is resized to 256×256 for comparison with other methods, while the experiment on editing wild identities uses the frames with resolution 512×512.

2.3. Optimization

The codebase for all these experiments is built upon PyTorch [17]. We randomly sample $T = 5$ frames for both source and driving sequences during training. The Reconstructor R is separately trained with the same setting as depicted by Deng et al. [6] and the landmarks obtained by the widely used framework [2] are used as the pseudo labels. The AdamW [15] optimizer is adopted with $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and the OneCycle [24] learning rate policy is used with the base learning rate 1×10^{-4} and the maximum learning rate 4×10^{-4}. The batch size is 64 over 8×32G NVIDIA Tesla V100 GPUs, in which 8 frames will be dispatched to each GPU. All these models are trained with 100 epochs for a fair comparison. The loss wights are $\lambda_p = 10, \lambda_v = 10, \lambda_k = 1, \lambda_e = 10, \lambda_w = 5$, and $\lambda_G = 1$.

2.4. Metrics

L1 distance (L1). To evaluate the reconstruction ability of models, we compute the mean L_1 distance, between the synthesized and driving frames. The values of RGB channels are normalized to $[0, 1]$.

Peak Signal-to-Noise Ratio (PSNR). The PSNR is used to measure the image reconstruction quality. PSNR is the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation [28].

Multi-Scale Structural Similarity (MS-SSIM). SSIM measures the structural similarity between two image patches. MS-SSIM is a multi-scale version of SSIM that measures on multiple scales of the images. We only report the MS-SSIM scores as it's shown to be more correlated with human perceptions.

Table 1. The complexity comparison of the different methods.

Method	MACs (G)	Params. (M)	Runtime (ms/frame)
FOMM [21]	56.36	59.79	**15.73**
MRAA [22]	61.01	66.00	17.92
OSFV [26]	521.85	125.30	140.04
TPSMM [31]	142.20	85.10	30.37
LIA [27]	88.31	45.10	26.73
DaGAN [11]	74.29	92.77	28.07
Face2Face$^{\rho}$ [30]	**3.20**	**11.31**	21.81
PECHead	357.10	161.09	71.07

Fŕechet Inception Distance (FID). The FID [10, 19] is used to evaluate the photo-realism of the synthesized frames.

Fŕechet Video Distance (FVD). We calculate the FVD score [25] of generated sequences to reveal the video quality.

Average Keypoint Distance (AKD). The average keypoint distance (AKD) [20, 21] is adopted to measure the semantic consistency. The facial landmark detector [2] is used to detect landmarks of driving and synthesized images and then the average distance between the corresponding landmarks in two images is computed.

Cross-Identity Similarity (CSIM). The cross-identity similarity (CSIM) [7, 21, 22, 26] is widely used to evaluate the identity preservation for cross-identity video face reenactment. Following the common settings, the ArcFace [5] is used to extract the face embeddings of the source and driving images, and the cosine similarity is computed between the two embeddings.

Average Rotation Distance (ARD). The average rotation distance (ARD) [7] is used to measure the head pose preservation. The camera parameters from 3D face reconstruction [6] are used to compute the Euler angles corresponding to head poses in the driving and synthesized frames. Then, the average $l1$-distance of the Euler angles across all frames is determined, and we report these values in degrees. In the frontalization task, the average rotation error (ARE) is used to measure the Euler angle errors, where the ground truth is the ideal frontal pose, with zero Euler angles.

Facial Action Unit Hamming Distance (AUH). The facial action unit Hamming distance (AUH) [7] is used to measure the expression preservation. We use the `py-feat` [3] toolbox to extract the facial action units (AUs) from the driving and synthesized images, and the AUH is computed as the average Hamming distance between the AUs of the driving and synthesized images.

Table 2. Quantitative results for the cross-identity reenactment.

Methods	VoxCeleb2				TalkingHead-1KH			
	CSIM↑	ARD↓	AUH↓	FVD↓	CSIM↑	ARD↓	AUH↓	FVD↓
FOMM [21]	0.716	1.74	0.172	226.5	0.682	5.44	0.221	240.3
MRAA [22]	0.677	2.19	0.143	254.2	0.564	2.11	0.093	243.4
OSFV [26]	0.772	3.02	0.164	234.4	0.845	2.98	0.151	224.4
TPSMM [31]	0.632	2.22	0.144	221.2	0.731	1.76	0.094	241.9
LIA [27]	0.775	3.35	0.146	268.1	0.708	2.20	0.115	233.8
DaGAN [11]	0.698	2.16	0.177	217.7	0.701	5.56	0.204	240.6
Face2Face$^{\rho}$ [30]	0.779	2.75	0.125	260.2	0.802	1.35	0.111	230.4
PECHead	**0.797**	**1.59**	**0.134**	**210.0**	**0.899**	**0.79**	**0.085**	**216.7**

3. Additional Results

3.1. Complexity

Table 1 shows the complexities of our method and other state-of-the-art methods. The number of MACs and parameters are calculated under the frame resolution of 256×256, and we utilize the `THOP` [33] toolbox to calculate them with pure PyTorch [17] implementations and no further optimization. The runtime is measured on a single NVIDIA RTX 2080Ti GPU. Compared with the real-time oriented method Face2Face$^{\rho}$ [30] and early works, *e.g.*, FOMM [21] and MRAA [22], our method has higher complexity, but significantly better performance. The OSFV [26] utilizes 3D convolution networks to extract the 3D learned landmarks and estimate the motion of the 3D learned landmarks, leading to a larger complexity, but the performance is inferior to our model. Among all these methods, our method has the best quantitative and qualitative performance but with acceptable complexity.

3.2. Video Face Reconstruction

Additional results of the video face reconstruction are shown in Figure 2. The top two rows are from the TalkingHead-1KH [26] dataset, the third and fourth rows are from the CelebV-HQ [32] dataset, and the last six rows are from the VFHQ [29] dataset.

3.3. Video Face Reenactment

The results of the video face reenactment are shown in Figure 3. Table 2 shows the quantitative results of the cross-identity reenactment. The top three rows are from the TalkingHead-1KH [26] dataset, the fourth through sixth rows are from the CelebV-HQ [32] dataset, and the last six rows are from the VFHQ [29] dataset.

3.4. Pose and Expression Editing

The frontalization results are shown in Figure 4. The expression editing results are shown in Figure 5, where the models are required to transfer the target expression to the source face. All the samples are from the VFHQ [29] dataset.

3.5. Reenactment on Wild Identities

To validate the generalization ability, we present more results of reenactment on wild identities, shown in Figure 6, where the source identity images are downloaded from the Internet, and the driving frames are from the VFHQ [29] dataset.

4. Free Editing on Wild Identities

The proposed method **PECHead** can also be used for free editing on wild identities. Additional results of pose editing on wild identities are shown in Figure 7. The results of expression editing on wild identities are shown in Figure 8.

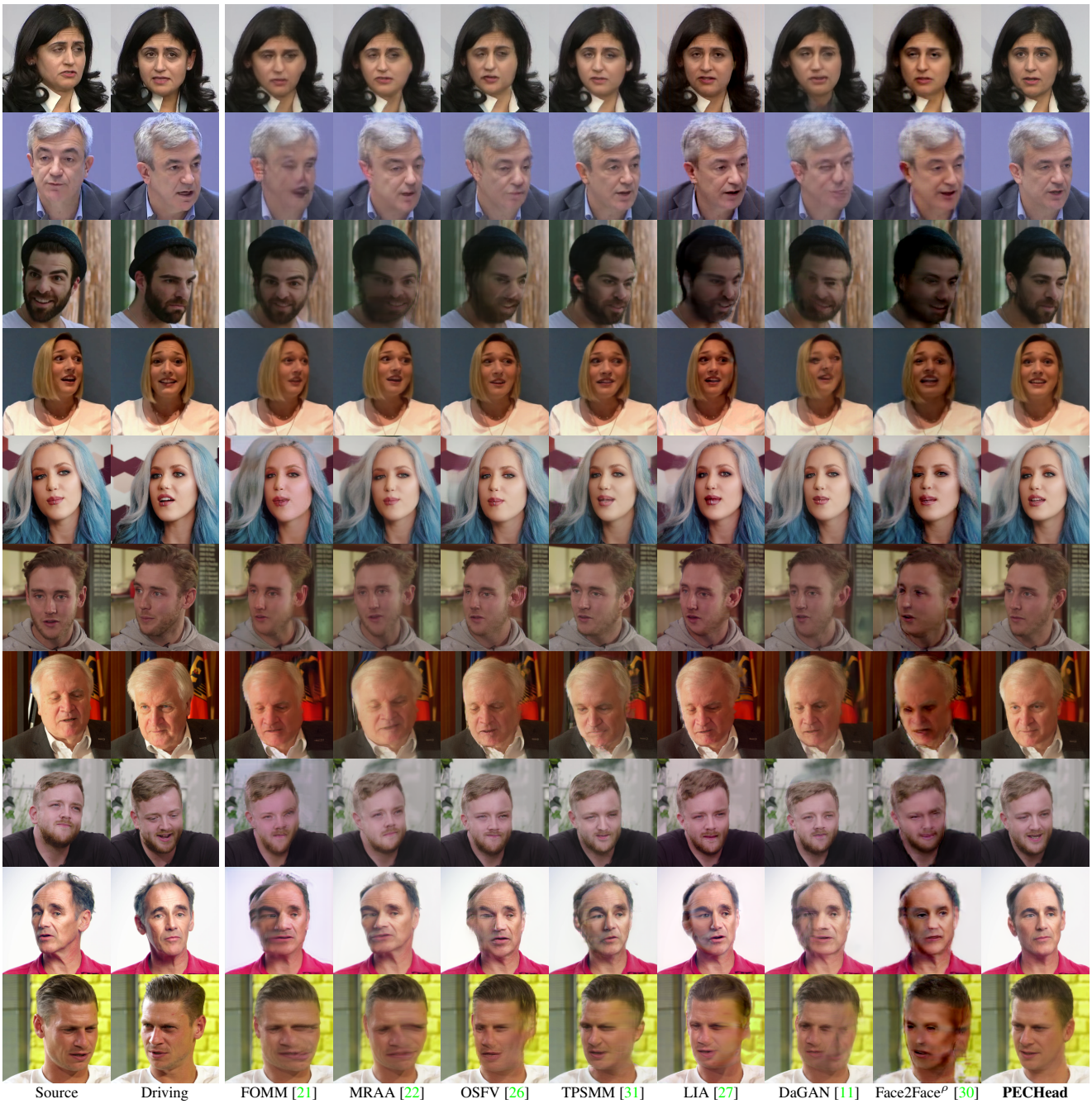


Figure 2. Comparison of same-identity video reconstruction results obtained by the proposed method and other state-of-the-art approaches.

Figure 3. Comparison of cross-identity face reenactment results obtained by the proposed method and other approaches.

Figure 4. Frontalization results of proposed method.

Figure 5. Expression editing results of proposed method.

Figure 6. Comparison of cross-identity face reenactment on wild identities.

Figure 7. Head pose free editing on wild identities.

Figure 8. Facial expression free editing on wild identities.

References

[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 1

[2] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 1021–1030, 2017. 3

[3] Jin Hyun Cheong, Tiankang Xie, Sophie Byrne, and Luke J Chang. Py-feat: Python facial expression analysis toolbox. *arXiv preprint arXiv:2104.03509*, 2021. 3

[4] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622*, 2018. 1

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 3

[6] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[7] Michail Christos Doukas, Stefanos Zafeiriou, and Viktoriia Sharmanska. Headgan: One-shot neural head synthesis and editing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14398–14407, 2021. 3

[8] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82. IEEE, 2018. 1

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[10] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017. 3

[11] Fa-Ting Hong, Longhao Zhang, Li Shen, and Dan Xu. Depth-aware generative adversarial network for talking head video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3397–3406, 2022. 1, 3, 4, 5, 6, 9

[12] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 1

[13] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 1

[14] Jae Hyun Lim and Jong Chul Ye. Geometric gan. *arXiv preprint arXiv:1705.02894*, 2017. 1

[15] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[16] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018. 1

[17] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 3, 4

[18] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? *Neural computation*, 16(5):1063–1076, 2004. 1

[19] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.2.1. 3

[20] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2377–2386, 2019. 3

[21] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. First order motion model for image animation. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 3, 4, 5, 6, 9

[22] Aliaksandr Siarohin, Oliver J Woodford, Jian Ren, Menglei Chai, and Sergey Tulyakov. Motion representations for articulated animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13653–13662, 2021. 1, 3, 4, 5, 6, 9

[23] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[24] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 3

[25] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018. 3

[26] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10039–10049, 2021. 1, 3, 4, 5, 6, 9

[27] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. Latent image animator: Learning to animate images via latent space navigation. *arXiv preprint arXiv:2203.09043*, 2022. 3, 4, 5, 6, 9

[28] Wikipedia contributors. Peak signal-to-noise ratio — Wikipedia, the free encyclopedia, 2022. [Online; accessed 18-November-2022]. 3

[29] Liangbin Xie, Xintao Wang, Honglun Zhang, Chao Dong, and Ying Shan. Vfhq: A high-quality dataset and benchmark for video face super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 657–666, 2022. 1, 3, 4

[30] Kewei Yang, Kang Chen, Yuan-Chen Guo, Daoliang Guo, Song-Hai Zhang, and Weidong Zhang. Face2face$^{\rho}$: Real-time high-resolution one-shot face reenactment. In *European conference on computer vision*. Springer, 2022. 3, 4, 5, 6, 9

[31] Jian Zhao and Hui Zhang. Thin-plate spline motion model for image animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3657–3666, 2022. 1, 3, 4, 5, 6, 9

[32] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. Celebv-hq: A large-scale video facial attributes dataset. *arXiv preprint arXiv:2207.12393*, 2022. 1, 4

[33] Ligeng Zhu. THOP: PyTorch-OpCounter. `https://github.com/Lyken17/pytorch-OpCounter.git`, November 2018. 4