Implicit Motion Function - Supplementary Material

Yue Gao Jiahao Li Lei Chu Yan Lu
Microsoft Research
{yuegao, li.jiahao, lei.chu, yanlu}@microsoft.com

1. Overview

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.

2. Loss Functions

Our model is trained using the self-supervised learning pipeline with a reconstruction task.

Pixel-wise Loss $\mathcal{L}_p$. The pixel-wise loss is employed to ensure the synthesis frame $\hat{x}_c$ is similar to the current frame x_c.

$$\mathcal{L}_p = \mathbb{E}[\| \hat{x}_c - x_c \|_1]. \tag{1}$$

Perceptual Loss $\mathcal{L}_v$. Similar to the existing methods [3, 18, 20, 25], we use a pre-trained VGG [8] to guarantee consistency of high level characteristics between the current frame x_c and reconstructed frame $\hat{x}_c$.

$$\mathcal{L}_v = \mathbb{E}[\sum_i \sum_j \| \text{VGG}^j(\hat{x}_c^i) - \text{VGG}^j(x_c^i) \|_1], \tag{2}$$

where i represents that the frame is downsampled i times, and j is the layer index of the VGG. We employ settings consistent with existing methods [3, 18, 20, 25], i.e. $i \in [0, 3]$ and $j \in [0, 4]$.

GAN Loss $\mathcal{L}_G, \mathcal{L}_D$. To make the synthesized frames realistic, we adopt the hinge adversarial loss [11], and two different scale patch discriminator is used for better performance [7].

$$\begin{aligned} \mathcal{L}_G &= -\mathbb{E}[D(\hat{x}_c)], \\ \mathcal{L}_D &= \mathbb{E}[\max(0, 1 - D(x_c) + \max(0, 1 + D(\hat{x}_c)]. \end{aligned} \tag{3}$$

Full Objective Function. The total loss of the generation step is formulated as:

$$L_G = \lambda_p \mathcal{L}_p + \lambda_v \mathcal{L}_v + \lambda_G \mathcal{L}_G, \tag{4}$$

where λ_p, λ_v and λ_G are the weights of loss functions, which equals to 10, 10 and 1, respectively. The loss of the discrimination step is formulated as $L_D = \mathcal{L}_D$. We follow the standard GAN practice [7] during training.

3. Implementation Details

3.1. Model Details

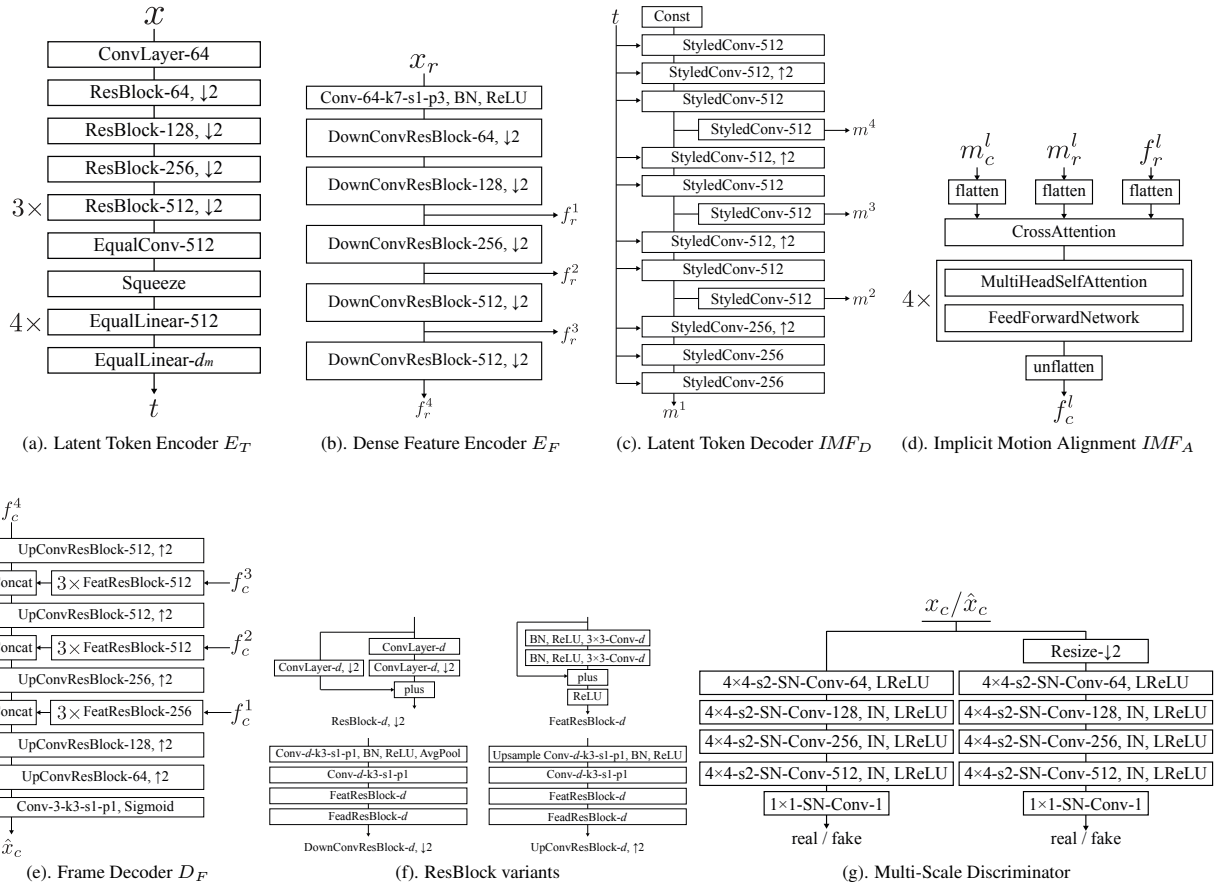
The details of the model structures and sub-modules are shown in Figure 1. Our encoder-decoder framework mainly contains four parts, the *dense feature encoder* E_F, the *latent token encoder* E_T, the *implicit motion function* (IMF) and the *frame decoder* D_F, where the IMF is composed of the *latent token decoder* IMF_D and *implicit motion alignment* IMF_A . The ConvLayer [10] block and StyledConv [10] block are directly adopted from the StyleGAN2-pytorch [16] implementation. The E_T is composed of several ResBlocks [4] and downsample blocks, and a multi-layer perceptron (MLP) is appended to the last, to finally obtain the latent token representation. The *latent token decoder* IMF_D is implemented with a StyleGAN2 [10] generator, and the latent token t_c is injected into the layers using the style modulation operation. For the *implicit motion alignment* IMF_A process, it can be formulated as:

$$\begin{aligned} V' &= \text{Attention}(Q, K, V), \\ &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V. \end{aligned} \tag{5}$$

With the aligned values V', we can further refine them using multi-head self-attention and feed-forward network-based Transformer blocks [19]. In this work, we use 4 stacked transformer decoder blocks.

$$\begin{aligned} \text{TransformerBlock}(x) &= \text{FFN}\Big(\text{MultiHeadSA}(x)\Big), \\ \text{head}_i &= \text{Attention}(xW_{Qi}, xW_{Ki}, xW_{Vi}), \\ \text{MultiHeadSA}(x) &= \text{Cat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W_O, \\ \text{FFN}(x) &= \text{GELU}\Big(\text{LN}(x)W_1 + b_1\Big)W_2 + b_2, \\ \text{GELU}(x) &= x \cdot \Phi(x), \end{aligned} \tag{6}$$

where the SA is the self-attention, which takes the output from the previous block, Cat is the concatenation operation, FFN is the feed-forward network, LN is the Layer Normalization [1], GELU [5] is utilized as the activation function and $\Phi(x)$ is the cumulative distribution function for Gaussian distribution.



(a). Latent Token Encoder E_T (b). Dense Feature Encoder E_F (c). Latent Token Decoder IMF_D (d). Implicit Motion Alignment IMF_A

(e). Frame Decoder D_F (f). ResBlock variants (g). Multi-Scale Discriminator

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

3.2. Datasets Details

Two talking head datasets, *i.e.*, CelebV-HQ [26], and VFHQ [23], are used in this paper. Apart from facial datasets, we also utilize three general datasets, *i.e.*, Flower, Wavecloth, and Foliage. For more details about these three datasets, please contact us. The resolution of frames is resized to 256×256 for all the experiments.

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 VFHQ, we follow the approaches used in [23] 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.

GeneralVideo. We adopt a large-scale text-video dataset. Please contact us for the detail information of the dataset. We use the words "flower", "wavecloth" and "foliage" to filter the captions to obtain the sub-datasets Flower, Wavecloth, and Foliage. Flower sub-dataset contains 50,837 training videos and 89 validation videos. Wavecloth sub-dataset contains 47,707 training videos and 23 validation videos. Foliage sub-dataset contains 1,003 training videos and 118 validation videos. For the foliage experiments, we first pretrain all the methods on the flower dataset, and then fine-tune on the foliage dataset.

3.3. Optimization

The codebase for all these experiments is built upon PyTorch [14]. The Adam [12] optimizer is adopted with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, and the learning rate policy is set to 2×10^{-4}. The batch size is 64 over 8×32G NVIDIA Tesla V100 GPUs, in which 8 training samples will be

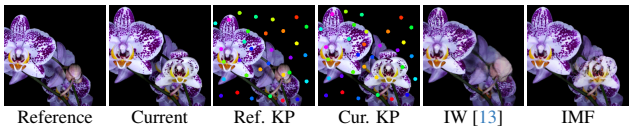


Figure 2. Visualization of IW keypoints and results of IW and IMF.

Figure 3. Results of using different reference frames.

dispatched to each GPU. All these models are trained until convergence for fair comparisons. The loss wights are $\lambda_p = 10, \lambda_v = 10$, and $\lambda_G = 1$.

3.4. 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 [22].

Structural Similarity (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 is shown to be more correlated with human perceptions.

Fŕechet Inception Distance (FID). The FID [6, 15] is used to evaluate the photo-realism of the synthesized frames.

Learned Perceptual Image Patch Similarity (LPIPS). The LPIPS [24] metric is an advanced method for assessing the perceptual similarity between images. It utilizes deep neural networks to compare image patches, focusing on human perceptual similarity rather than pixel-level differences.

4. Additional Analysis

4.1. Limitations of Explicit Representation

We visualized the results of IW [13], which uses attention but is limited by its use of explicit keypoints from FOMM, restricting its applicability and generalizability. By contrast, our IMF, leveraging low-dimensional tokens in latent space without physical coordinate constraints, enhances motion representation sparsity and adaptability, thereby improving video modeling. It ensures semantic integrity while maintaining sparsity, enhancing the generalizability and ability to represent a wider range of object movements and behaviors, as seen in flower scenarios (Fig. 2).

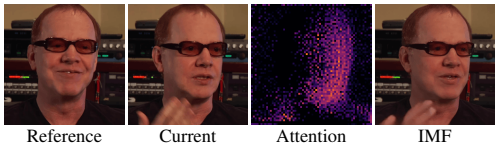


Figure 4. Visualization of the attention map.

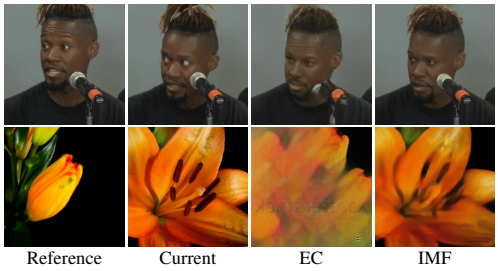


Figure 5. Results of encoder-centric (EC) model and IMF.

4.2. Choosing the Reference Frame

Reference frame selection greatly impacts reconstruction quality, with closer frames typically improving results, shown in Fig. 3. Currently, we follow previous papers and also use the first frame for fair comparison. But an adaptive reference frame selection is more reasonable and we will research it in the future.

4.3. Visualization of the Attention Map

We add attention map visualization in Fig. 4. It shows our IMF effectively discerns the correlation between two frames and identifies new elements like a hand.

4.4. Encoder-centric (EC) or Decoder-centric (DC)

Fig. 5 compares an encoder-centric (EC) model and our DC model, IMF. It shows EC has poor preservation of face identity and non-face element, and limited generalizability to general data. Traditional EC methods require complex encoder designs and meticulous tuning during training phases, while our DC approach places correlation modeling within the decoder, simplifying model design and avoiding complex turning. Our IMF design also aligns with successful DC-based Large Language Models (LLMs), and it is expected to achieve similar enhancements in video modeling. We will provide an expanded discussion on EC and DC in the revision. Theoretical support comes from Wyner-Ziv coding, indicating that DC frameworks can match EC ones in rate.

5. Additional Results

5.1. Talking Head Video Reconstruction

Additional results of the talking head video reconstruction are shown in Figure 6. We can see that our proposed IMF can faithfully reconstruct the frames including no-facial contents, such as the hand and the microphone.

5.2. General Video Reconstruction

The results of the general video reconstruction are shown in Figure 7. For general video frames, the proposed method IMF can reconstruct the general data with high-fidelity, which faithfully models the subtle movement of the foliage, the blooming of the flowers, as well as the waving cloth.

5.3. Token Editing for Head Pose

We compare our results with the previous SOTA method PECHead [3], as they explicit utilizing the 3DMM [2] face coefficients. All the samples are the in-the-wild images from the FFHQ [9] dataset.

Given Head Pose Editing. The results of token editing for head pose are shown in Figure 8, where the head pose are extracted from the first row "target pose".

Free Head Pose Editing. The results of free pose editing are shown in Figure 9, where the head pose are arbitrarily set.

5.4. Token Editing for Expression

The results of token editing for facial expression are shown in Figure 10, where the head pose are extracted from the first row "target exp.". All the samples are the in-the-wild images from the FFHQ [9] dataset.

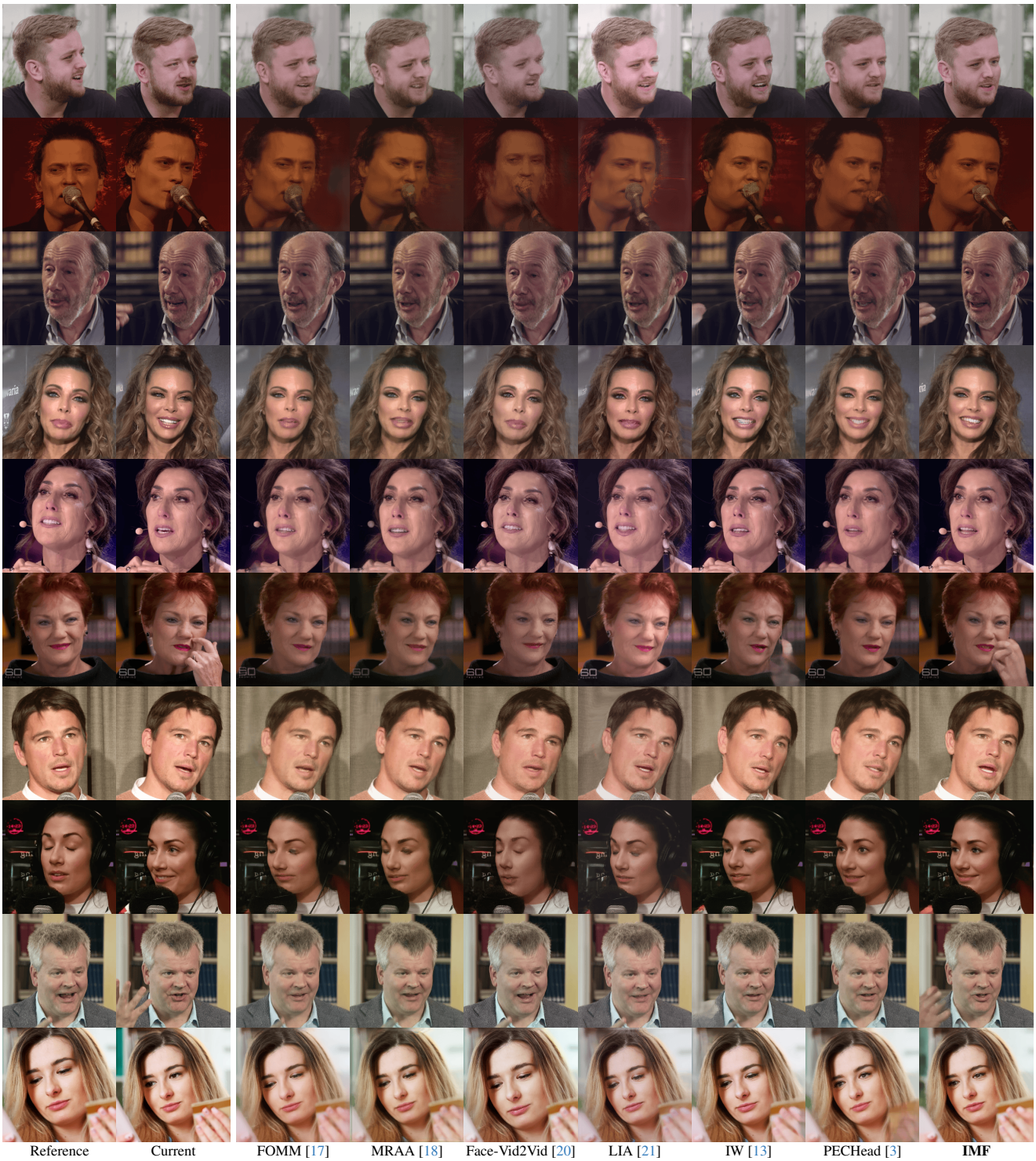


Figure 6. Comparison of talking head video reconstruction results obtained by the proposed method and previous SOTA approaches.

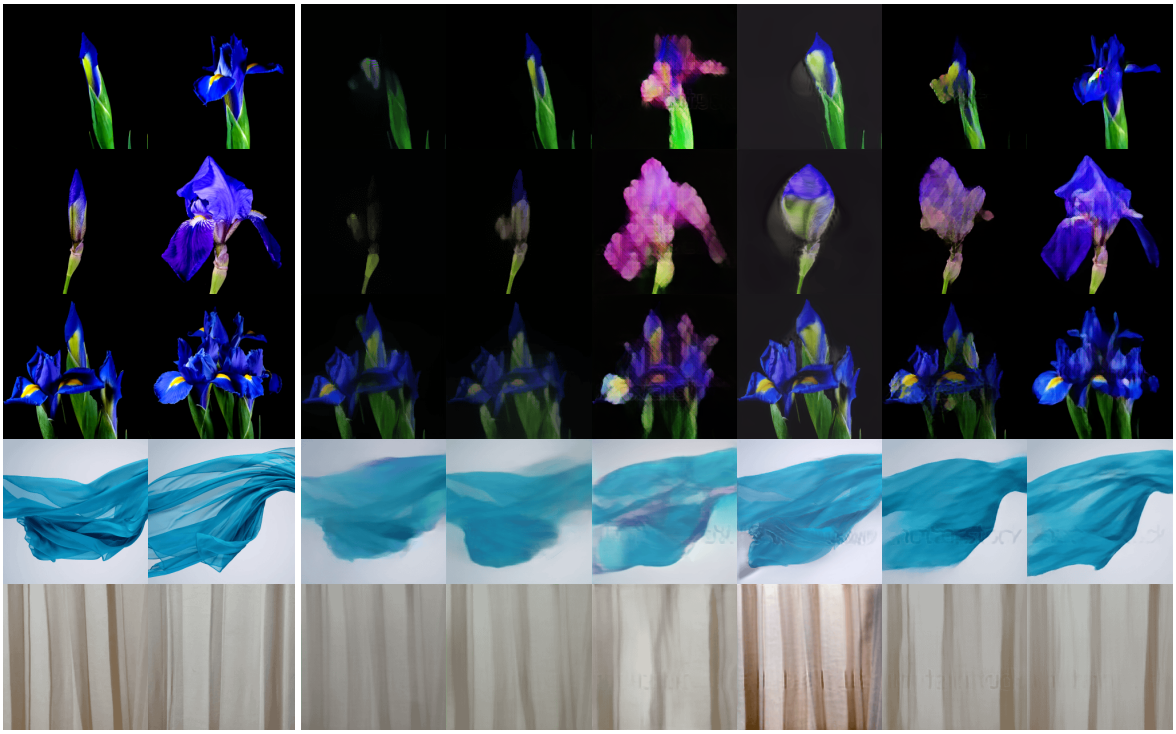


Figure 7. Comparison of talking head video reconstruction results obtained by the proposed method and previous SOTA approaches.

Figure 8. Head pose editing results of the proposed method IMF and the previous SOTA PECHead [3].

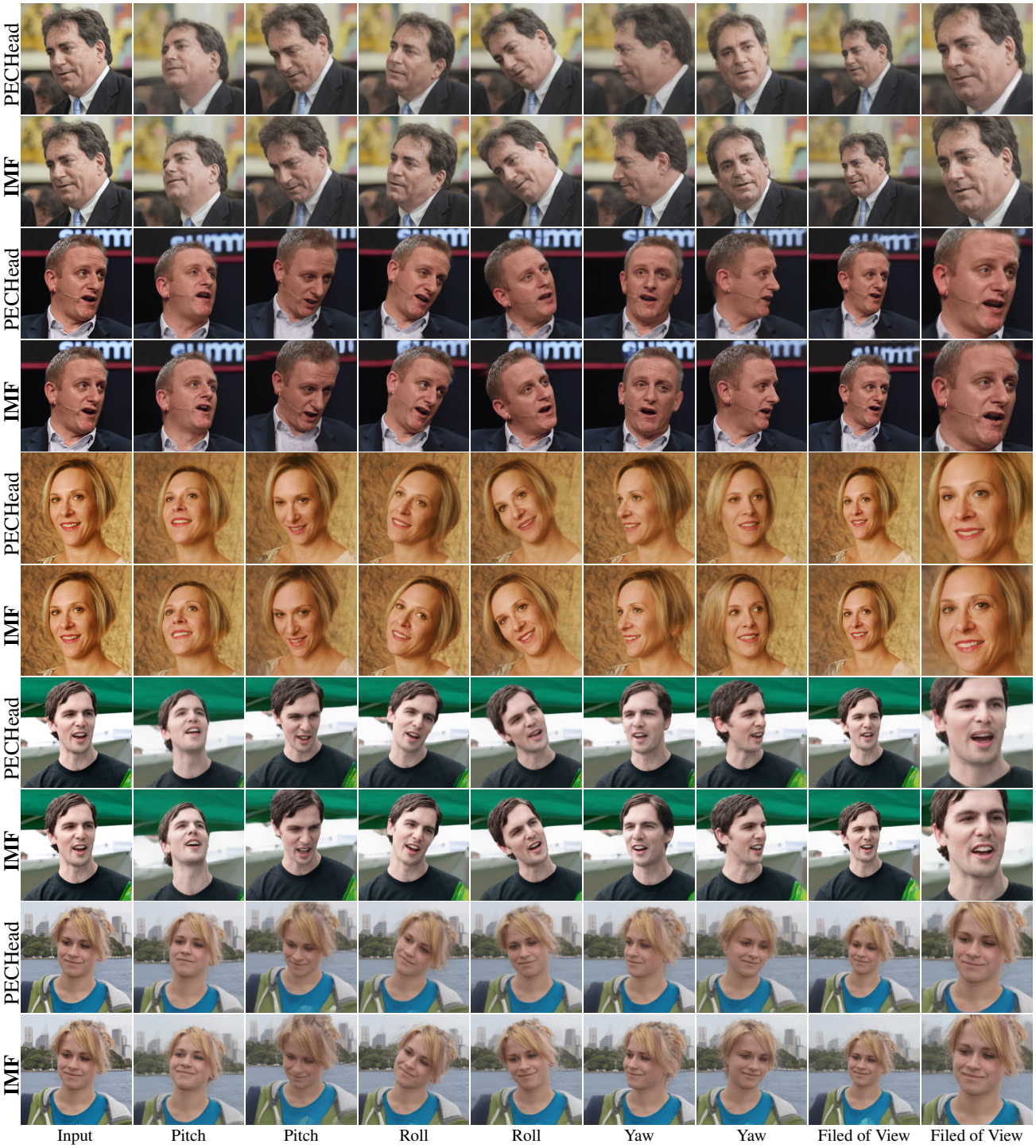


Figure 9. Comparison of talking head pose free editing results obtained by the proposed method and the state-of-the-art approach PECHead [3].

Figure 10. Expression editing results of the proposed method IMF and the previous SOTA PECHead [3].

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1

[2] 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. 4

[3] Yue Gao, Yuan Zhou, Jinglu Wang, Xiao Li, Xiang Ming, and Yan Lu. High-fidelity and freely controllable talking head video generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5609–5619, 2023. 1, 4, 5, 7, 8, 9

[4] 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

[5] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 1

[6] 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

[7] 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

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 1

[9] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 4

[10] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 1

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

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[13] Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. Implicit warping for animation with image sets. *Advances in Neural Information Processing Systems*, 35:22438–22450, 2022. 3, 5, 6

[14] 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. 2

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

[16] Kim Seonghyeon. stylegan2-pytorch. https://github.com/rosinality/stylegan2-pytorch, 2023. 1

[17] 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. 5, 6

[18] 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, 5, 6

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[20] 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, 5

[21] 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. 5, 6

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

[23] 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. 2

[24] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 3

[25] 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, 6

[26] 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. 2