EMOPortraits: Emotion-enhanced Multimodal One-shot Head Avatars

Supplementary Material

1. Main model implementation details

1.1. Full List of Findings

Beyond the core findings highlighted in the main text, this section outlines additional key distinctions that set our work apart from the MegaPortraits model [3].

Model Architecture. Our main model's structure, depicted in Figure 4 of the main text, excluding the speech-driven component, shares a conceptual resemblance with that of MegaPortraits [3]. However, we have implemented several architecture alterations. First is the reduction in the dimensionality of the latent expression descriptors from 512 to 128, a change detailed in Section 4.1. This reduction enhances the efficiency of the model without compromising the quality of expression representation. Additionally, we have made comprehensive modifications to the architecture and size of the model's main components to optimize model's performance. These modifications are visually represented in Fig. 1, and the detailed architectures can be explored in Fig. 8.

Dropout. We have incorporated dropout as a last layer of $\mathbf{E}_{\text{motion}}$ that predicts the expression descriptors ($\mathbf{z}$) in our model. This implementation serves to improve $\mathbf{E}_{motion}$'s capability to construct a more nuanced and robust latent representation of facial expressions, and also aids in preventing overfitting. By ensuring that the model does not become overly reliant on any element of the latent vector, we achieve a more generalized and versatile expression representation capability, essential for dealing with a wide range of facial motions.

Enhanced Loss Functions. Beyond new loss functions introduced in the main text, such as the canonical volume loss (outlined in Section 4.2) and the source-driver mismatch loss (described in Section 4.3), we have also integrated the $\mathcal{L}_{\text{head}}$ loss, as mentioned in Section 1.3. This specific loss function plays a noticeable role in the precise predicting of facial regions critical for emotional expression, particularly the eyes and mouth. Additionally, it addresses the previously noted challenges in accurately generating ears. The integration of this loss underscores our model's attention to detail and commitment to achieving a high degree of realism in facial expression synthesis.

1.2. Implementation Details

Data Preparation. Our data preparation approach for the VoxCeleb2 (VC2) dataset [1] follows the protocol established in the original model [3]. For our novel FEED dataset, we cropped frames around the face region and resized them. The dataset subset used in our experiments, detailed in Table 1, includes ”Winks,” ”Tongue Emotion,” and ”Extreme & Asymmetric Emotion.” Notably, during training, we did not exploit the multi-view nature of the FEED dataset. Specifically, in each iteration, both the source and driver images were selected from the same single-camera video. Both datasets were employed for training and evaluating our model at a resolution of 512 × 512.

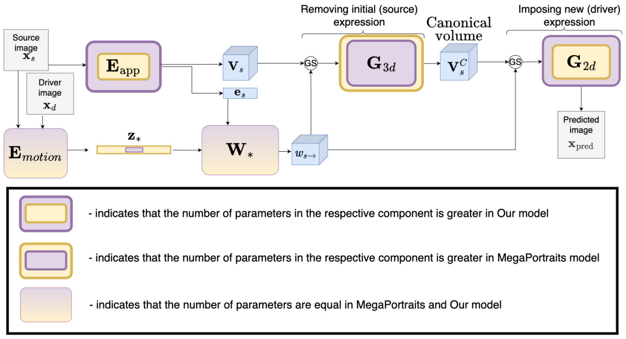
Removing initial (source) expression
Canonical volume
Imposing new (driver) expression
Source image $\mathbf{x}_s$
Driver image $\mathbf{x}_d$
$\mathbf{E}_{\text{app}}$
$\mathbf{V}_s$
$\mathbf{e}_s$
GS
$\mathbf{G}_{3d}$
$\mathbf{V}_s^C$
GS
$\mathbf{G}_{2d}$
Predicted image $\mathbf{x}_{\text{pred}}$
$\mathbf{E}_{motion}$
$\mathbf{z}_*$
$\mathbf{W}_*$
$w_{s\to}$
- indicates that the number of parameters in the respective component is greater in Our model
- indicates that the number of parameters in the respective component is greater in MegaPortraits model
- indicates that the number of parameters are equal in MegaPortraits and Our model

Figure 1. Comparison of our model's scheme with the MegaPortraits scheme, showing the relative sizes of individual components.

Training Details. Our framework was trained on 8 Nvidia Tesla A100 GPUs for 250,000 iterations, using a batch size of 2 per GPU (16 in total). The training data consisted of a mixture of 75% VC2 examples and 25% FEED examples. Every second iteration involved a batch comprising one image pair from VC2 and one from the FEED dataset. This sampling strategy, integrating image pairs from both datasets, proved more effective than using separate batches from each dataset. By employing contrastive losses where positive and negative pairs spanned both datasets, we mitigated overfitting risks associated with the limited identity variety in the FEED dataset. This approach also facilitated the asymmetric emotion translation to unseen identities.

1.3. Used losses

Photometrical losses. These are key to aligning the motion and appearance of the predicted image ($\hat{\mathbf{x}}_{s\to d}$) with the ground truth image ($\mathbf{x}_d$). To achieve this, we use three distinct pre-trained networks:

- *VGG19* [12] (ILSVRC/ImageNet [2] Trained): This helps in matching the overall content of the images.
- *VGGFace* [9] (Face Recognition Focused): Essential for aligning facial features accurately.
- *Gaze Direction Based on VGG16* [4]: Specifically trained to emulate a top-notch gaze detection system, ensuring precise gaze direction matching.

We measure the similarity by calculating the L1 distance between the feature maps of both the predicted and ground truth images, utilizing all these networks. Additionally, we employ face masks for the eyes, mouth, and ears (sourced from FaceParsing network [16]), focusing our model on these critical head areas. Then we use these masks to match mentioned head regions on the predicted and the ground truth images using L1 loss between pixel corresponded to a specific region. The final photometric loss combines these individual perceptual losses, formulated as:

$$\mathcal{L}_{\text{pho}} = w_{\text{IN}}\mathcal{L}_{\text{IN}} + w_{\text{face}}\mathcal{L}_{\text{face}} + w_{\text{gaze}}\mathcal{L}_{\text{gaze}} + \mathcal{L}_{\text{head}}.$$

Here, $\mathcal{L}_{\text{head}}$ further breaks down into:

$$\mathcal{L}_{\text{head}} = w_{\text{eyes}}\mathcal{L}_{\text{eyes}} + w_{\text{mouth}}\mathcal{L}_{\text{mouth}} + w_{\text{ears}}\mathcal{L}_{\text{ears}}.$$

Self-supervised losses. As detailed in Section 5, we trained our expression descriptors using a modified large margin cosine loss (CosFace) [13], denoted as $\mathcal{L}_{\text{cos}}$ and presented in Equation 5. This approach is similar to the one employed by the authors of MegaPortraits [3]. Additionally, we introduced two more losses. The source-driver mismatch loss $\mathcal{L}_{\text{sdm}}$ Equation 3 (described in Section 4.3), which directly influences the expression's latent space. This loss is pivotal in eliminating identity information from the expression descriptor $\mathbf{z}_i$ and in preventing overfitting, especially in the context of our extremely imbalanced dataset. The combination of these two losses forms our latent space loss:

$$\mathcal{L}_{\text{lat}} = w_{\text{cos}}\mathcal{L}_{\text{cos}} + w_{\text{sdm}}\mathcal{L}_{\text{sdm}}.$$

The second additional self-supervised loss that enhances the disentanglement of identity and expression is the canonical volume loss $\mathcal{L}_{\text{CV}}$ (described in Section 4.2 and Equation 2). This loss functions to extract expression information from the canonical volume, thereby reducing the overlap of information contained in $\mathbf{V}_i^C$ and $\mathbf{z}_i$.

Adversarial losses. To ensure the predicted images look realistic, adversarial losses are computed using the same predicted and reference images. We follow [3] and train a multi-scale patch discriminator [20] with a hinge adversarial loss. To boost training stability, a standard feature-matching loss is also included ([14]). The GAN loss for the generator is expressed as:

$$\mathcal{L}_{\text{GAN}} = w_{\text{adv}}\mathcal{L}_{\text{adv}} + w_{\text{FM}}\mathcal{L}_{\text{FM}}.$$

To conclude, the total loss which is used to train our model is the sum of individual losses:

$$\mathcal{L}_{\text{main}} = \mathcal{L}_{\text{pho}} + \mathcal{L}_{\text{lat}} + w_{\text{CV}}\mathcal{L}_{\text{CV}} + \mathcal{L}_{\text{GAN}}. \quad (1)$$

We utilized the AdamW optimizer [8] with a cosine learning rate schedule. The initial learning rate was gradually reduced from 2×10^{-4} to 1×10^{-6} over the training iterations. The hyperparameters for the losses were set as follows: $w_{\text{IN}} = 20$, $w_{\text{face}} = 10$, $w_{\text{gaze}} = 10$, $w_{\text{adv}} = 1$, $w_{\text{FM}} = 40$, $w_{\text{cos}} = 2$, $w_{\text{std}} = 1$ (increased to 10 for pairs from the FEED dataset), and $w_{\text{CV}} = 1$. Additionally, we set $s = 5$ and $m = 0.2$ in the cosine loss.

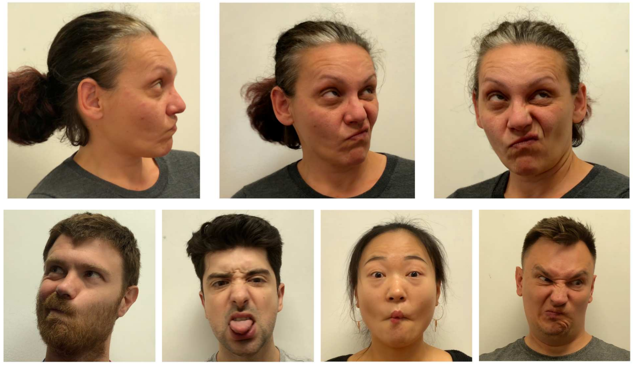

Figure 2. More selected examples from our FEED dataset.

1.4. Visual Comparison

Our choice of baseline methods, as outlined in our experiment section in the main text, was driven by two key factors. Firstly, these methods are prominent in the field of talking-head video generation using arbitrary identities, making them relevant benchmarks for our study. Secondly, the accessibility of their source code and pretrained model weights, either through public availability or provided by the authors for use with our test set, was a crucial consideration.

The setup for our visual comparison, as described in Section 6.1, was chosen based on specific criteria. We choose FFHQ images as source images due to the consistent clarity of facial features across the dataset, which is essential for accurate comparison. Additionally, it was critical to select identities that were not part of the training datasets for any of the methods used in comparison. This ensures that our comparisons are based on novel identities, providing a fair assessment of each method's generalization capabilities. This criterion was also applied in selecting driving identities from the MEAD and FEED datasets, which were not used in training by any of the compared methods.

To supplement the comparisons described in the main paper, we provide additional examples for each method (refer to Figure 5). As, for NOFA [17], the range of examples is limited due to the restricted number of inferred identities provided by the authors, we provide a second set of additional examples excluding NOFA [17] (see Figure 6). Furthermore, we include a visual comparison Figure 7 for our ablation study (see Table 4 in the main text).

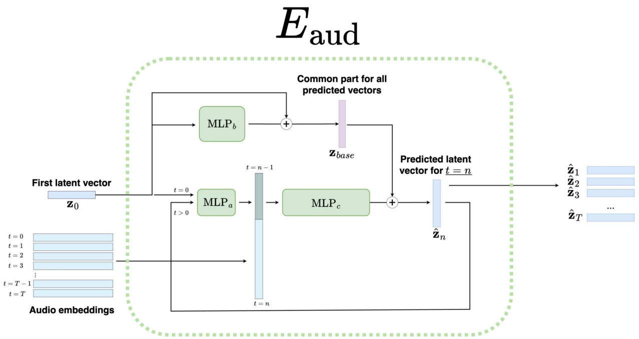

Figure 3. Comparison of our audio encoder used to predict latent expression vectors during speech-driven mode.

2. Speech-driven mode

2.1. Implementation details

In this section, we detail the workings of our audio encoder, $\mathbf{E}_{aud}$, as depicted in Fig. 4, for its application in speech-driven scenarios. The encoder, $\mathbf{E}_{aud}$, is designed for generating latent expression vectors, denoted as $\mathbf{z}$, from speech inputs. The scheme of the encoder presented on Fig. 3.

First, we use Whisper [10] model to retrieve audio embeddings from a raw audio clip containing speech. This step yields a series of T audio embedding vectors, where each vector is linked to a specific frame in the video clip. Next, we employ a multilayer perceptron (MLP), designated as MLP_b, to compute the base component of our latent expression vectors, $\mathbf{z}_{\text{base}}$. This base component encapsulates common facial features such as initial gaze direction and the pose of the upper facial region, as observed in the first frame of the input. Then, another MLP, MLP_a, is employed. It uses the previously calculated latent vector $\mathbf{z}$—initially which is $\mathbf{z}_0$ for $t = 0$ and $\hat{\mathbf{z}}_n$ for $t > 0$—to align the latent features of $\mathbf{z}$ with those of the audio embeddings and derive features that useful for final prediction. In the final step, after merging the relevant audio embedding with the output from MLP_a, the next network, MLP_c is utilized and responsible for computing a part of the vector which, when added to $\mathbf{z}_{\text{base}}$, forms the final latent expression vector $\hat{\mathbf{z}}_n$ for each frame.

2.2. Data Preparation

Similar to our primary model, we employ the VoxCeleb2 dataset [1] for training our audio encoder, $\mathbf{E}_{aud}$. During each training iteration, we randomly select an audio clip, varying in length from 50 to 200 frames. The corresponding audio segment and the initial frame of this clip are fed into $\mathbf{E}_{aud}$ as inputs. All subsequent frames from the clip are utilized as reference frames (ground truth) for training purposes. Both the training and evaluation processes are conducted using a resolution of 512×512.

2.3. Utilized Loss Functions

Training of $\mathbf{E}_{aud}$ incorporates three distinct types of loss functions:

Photometrical Identity Preservation Losses: To ensure the identity of the individual in the video is preserved, we apply an L1 loss between the predicted image (using output expression vectors from $\mathbf{E}_{aud}$, denoted as $\hat{\mathbf{x}}_n^{\text{aud}}$) and the ground truth image $\mathbf{x}_n$. Additionally, we implement a perceptual loss comparing facial features in these images using the *VGGFace* model [9]. The identity preservation loss is represented as:

$$\mathcal{L}_{\text{idt}} = w_{\text{L1}}\mathcal{L}_{\text{L1}} + w_{\text{face}}\mathcal{L}_{\text{face}}$$

Latent Mouth Movement Losses: For accurately translating mouth movements, we use an L1 loss focusing on the principal components related to mouth movements in $\hat{\mathbf{z}}_n$ and $\mathbf{z}_n$. This loss, detailed in Sec. 5.2, is denoted as $\mathcal{L}_{\text{PCA}}(\mathbf{z}_i, \mathbf{z}_j, n)$, with $n = 8$ in our experiments. A secondary L1 loss, $\mathcal{L}_{\text{vtr}}$, with a reduced weight, ensures a closer match between $\hat{\mathbf{z}}_n$ and $\mathbf{z}_n$ vectors.

$$\mathcal{L}_{\text{latent}} = w_{\text{PCA}}\mathcal{L}_{\text{PCA}} + w_{\text{vtr}}\mathcal{L}_{\text{vtr}}$$

Photometrical Lip Movement Losses: For enhanced translation of mouth movements, we employ the FaceParsing network [16] to generate masks for the upper lip, lower lip, and inner mouth regions in both the predicted ($\hat{\mathbf{x}}_n^{\text{aud}}$) and ground truth ($\mathbf{x}_n$) images. These masks are compared using the Binary Cross Entropy loss, $\mathcal{L}_{\text{BCE}}$. Additionally, we extract pixels corresponding to the mouth in both predicted and actual images, comparing them using an L1 loss, $\mathcal{L}_{\text{lips}}$:

$$\mathcal{L}_{\text{mouth}} = w_{\text{BCE}}\mathcal{L}_{\text{BCE}} + w_{\text{lips}}\mathcal{L}_{\text{lips}}$$

The overall loss function used to train our model is the cumulative sum of these individual losses:

$$\mathcal{L}_{\text{speech}} = \mathcal{L}_{\text{idt}} + \mathcal{L}_{\text{latent}} + \mathcal{L}_{\text{mouth}} \tag{2}$$

We utilized the AdamW optimizer [8] with a cosine learning rate schedule. The initial learning rate was gradually reduced from 1×10^{-4} to zero over the training iterations. The hyperparameters for the losses were set as follows: $w_{\text{L1}} = 10$, $w_{\text{face}} = 100$, $w_{\text{PCA}} = 200$, $w_{\text{vtr}} = 5$, $w_{\text{BCE}} = 5 * 10^3$, $w_{\text{lips}} = 5 * 10^5$.

3. Generating head rotations

To control the main model using head rotations generated from speech, we have developed a generative adversarial model. This model takes a speech recording as input and outputs a series of rotations, as depicted in Fig 4.

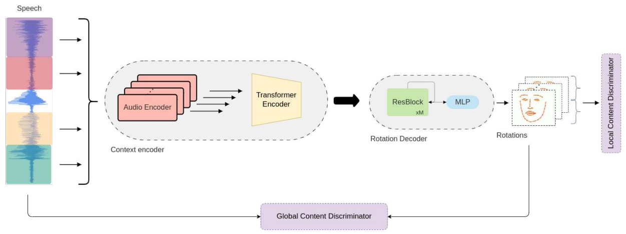

Figure 4. **Audio-to-rotations model**. The input signal is split into overlapping segments for the audio encoder, processed through a Transformer and a ResNet decoder. The resulting rotation sequence and audio input are then given to the Global Content Discriminator, while the Local Temporal Discriminator receives smaller segments of these rotations.

3.1. Rotation Representation

We represent the 3D rotations with six dimensions as described in [19]. This ensures the continuity of the representation, which is more suitable for learning. We also add an extra three parameters to predict the translation. Therefore, we can formulate a head transformation sequence X as a sequence of rotations and translation across T consecutive frames, $X \in R^{T\times 9}$ where each $X_t \in R^9$ is a vector representing the transformation from the reference frame. To map the 6D representation again to the 3D rotation group, we can use the following formula:

$$f_{GS}\left(\begin{bmatrix} | & | \\ a_1 & a_2 \\ | & | \end{bmatrix}\right) = \begin{bmatrix} | & | & | \\ b_1 & b_2 & b_3 \\ | & | & | \end{bmatrix} \tag{3}$$

$$b_i \left[\begin{cases} N(a_1) & \text{if } i = 1 \\ N(a_2 - (b_1 \cdot a_2)b_1) & \text{if } i = 2 \\ b_1 \times b_2 & \text{if } i = 3 \end{cases}\right]^T \tag{4}$$

Here $N(\cdot)$ denotes a normalization function and f_{GS} a Gram-Schmidt process. The model produces a 3x2 matrix with a_1, a_2 being its columns. The Gram-Schmidt process in Equation (4) produces the third column b_3 by taking the cross product of the two first columns b_1 and b_2, making it normal to the plane containing them. This process ensures that the resulting 3x3 matrix is orthogonal. The remaining three dimensions map directly to the translations by construction.

3.2. Generator

Our generator is split into two components: a context encoder and a head pose decoder. The context encoder merges an audio encoder, Whisper, with a transformer encoder for temporal analysis. This setup efficiently leverages existing audio embeddings from other parts of our system. The head pose is decoded using the encoder's hidden states, processed through ResNet layers and an MLP. This converts $R^{T\times H}$ to $R^{T\times 9}$, where H is the Transformer's hidden size.

3.3. Discriminators

To evaluate our generated rotations, we use two discriminators assessing local and global coherence.

Local Content Discriminator. Inspired by Isola et al. [6], we use a 1D temporal PatchGAN variant. This discriminator targets patch-level structures, classifying patches of N frames as real or fake. With the discriminator convolution spanning the entire sequence, averaging all responses yields the final judgment. We found N=8 optimizes frame-to-frame coherence.

Global Content Discriminator. This discriminator assesses the full sequence's coherence with the audio input. We encode the rotation sequence using 1D ResNet blocks and a global pooling layer, then concatenate it with the audio embeddings from Whisper. A final MLP layer determines if the sequence is authentic or generated.

3.4. Losses

To train the model, we use a weighted combination of different losses. The resulting loss is described as follows:

$$\mathcal{L}_{tot} = \lambda_{recons}\mathcal{L}_{recons} + \lambda_{adv}\mathcal{L}_{adv} + \lambda_{smooth}\mathcal{L}_{smooth} \tag{5}$$

Reconstruction loss $\mathcal{L}_{recons}$. Is a L_1 loss between predicted X and ground-truth Y head poses. The head poses can be separated into rotation r and translation t.

$$\mathcal{L}_{recons} = \sum_{i=1}^{T} |r_i - \hat{r}_i| + \sum_{i=1}^{T} |t_i - \hat{t}_i| \tag{6}$$

We then have two different coefficient factors λ_{rot} and λ_{trans} for the reconstruction of rotations and translations.

Smoothing loss $\mathcal{L}_{smooth}$. Acts as a regularization loss and ensures smoothness over consecutive frames.

$$\mathcal{L}_{smooth} = \sum_{i=2}^{T} |X_i - X_{i-1}| \tag{7}$$

Adversarial loss $\mathcal{L}_{adv}$. We adopt WGAN-GP [5] for improved stability of the training and for avoiding mode collapse. The discriminator and generator losses are as follows:

$$\begin{aligned} \mathcal{L}_D &= \mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)] \\ &+ \lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}}D(\hat{x})\|_2 - 1)^2] \end{aligned} \tag{8}$$

$$\mathcal{L}_G = -\mathbb{E}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] \tag{9}$$

$\mathbb{P}_{\hat{x}}$ is defined as a uniform sampling along straight lines between pairs of points sampled from the data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$. We set λ to 10 and update five times the discriminator for every single update of the generator as suggested in the original paper.

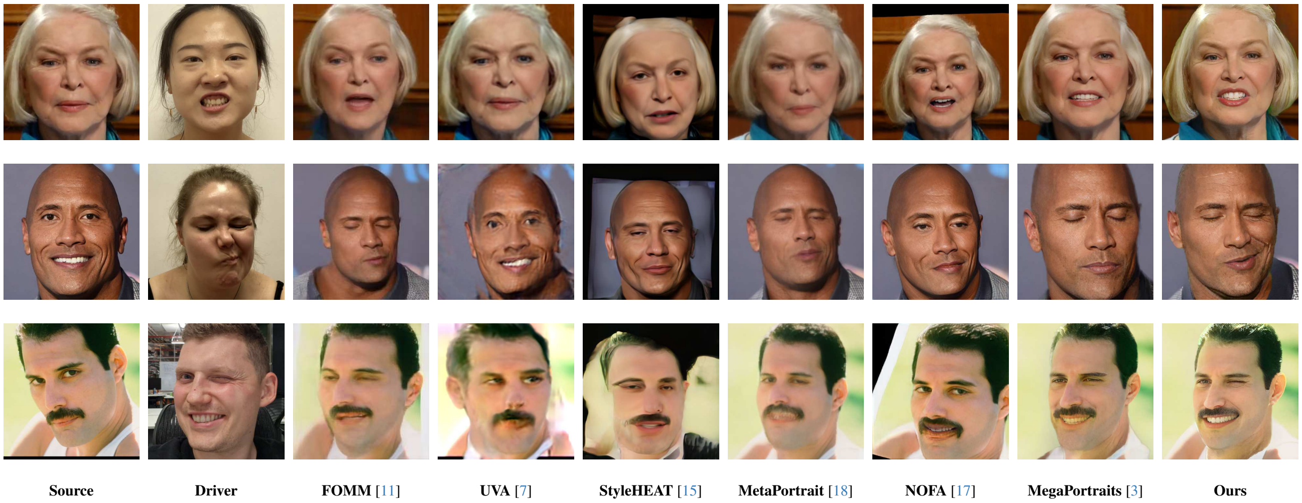

Figure 5. An additional qualitative comparison of head avatar systems in cross-reenactment scenario.

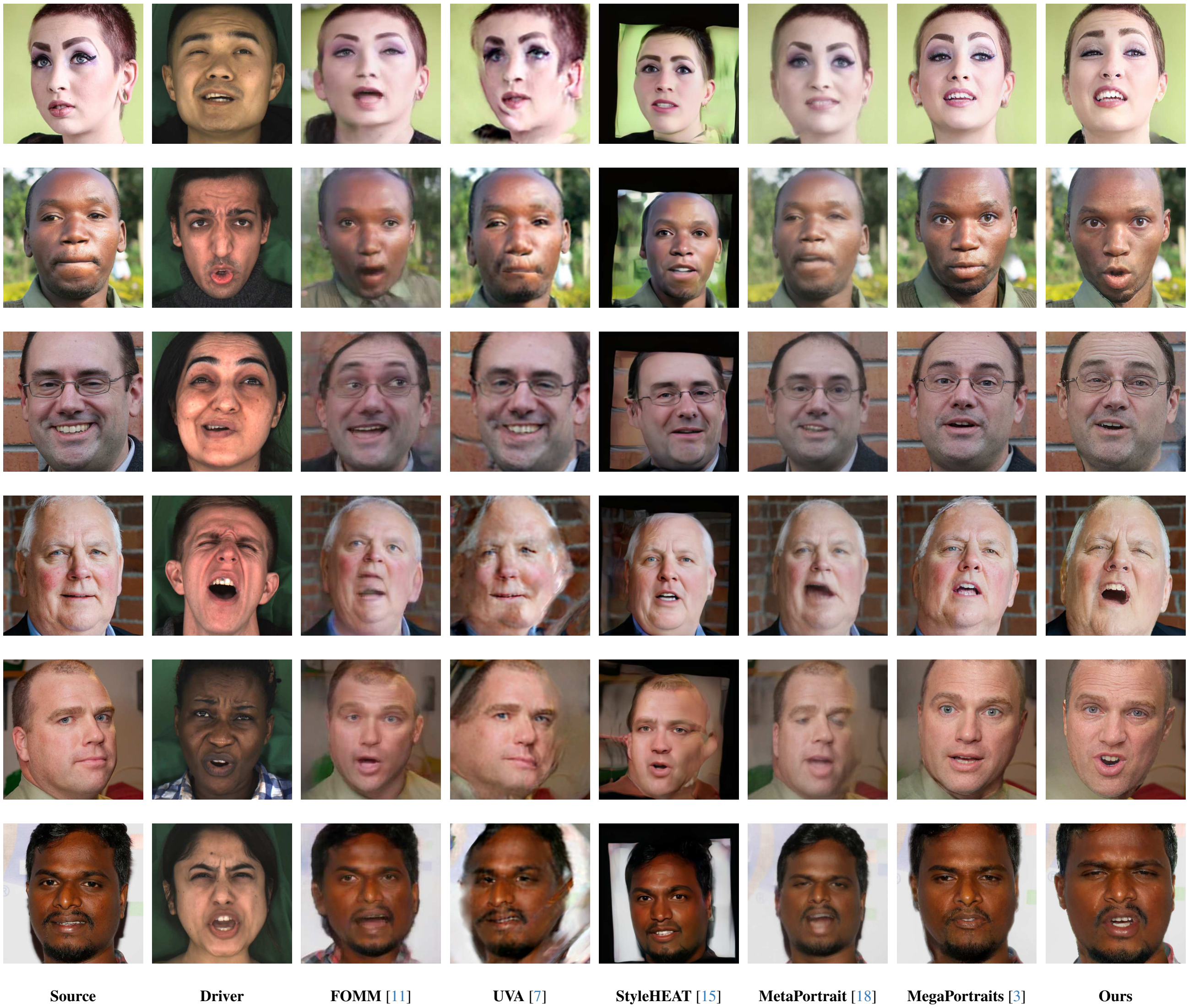

Figure 6. An additional qualitative comparison of head avatar systems in cross-reenactment scenario.

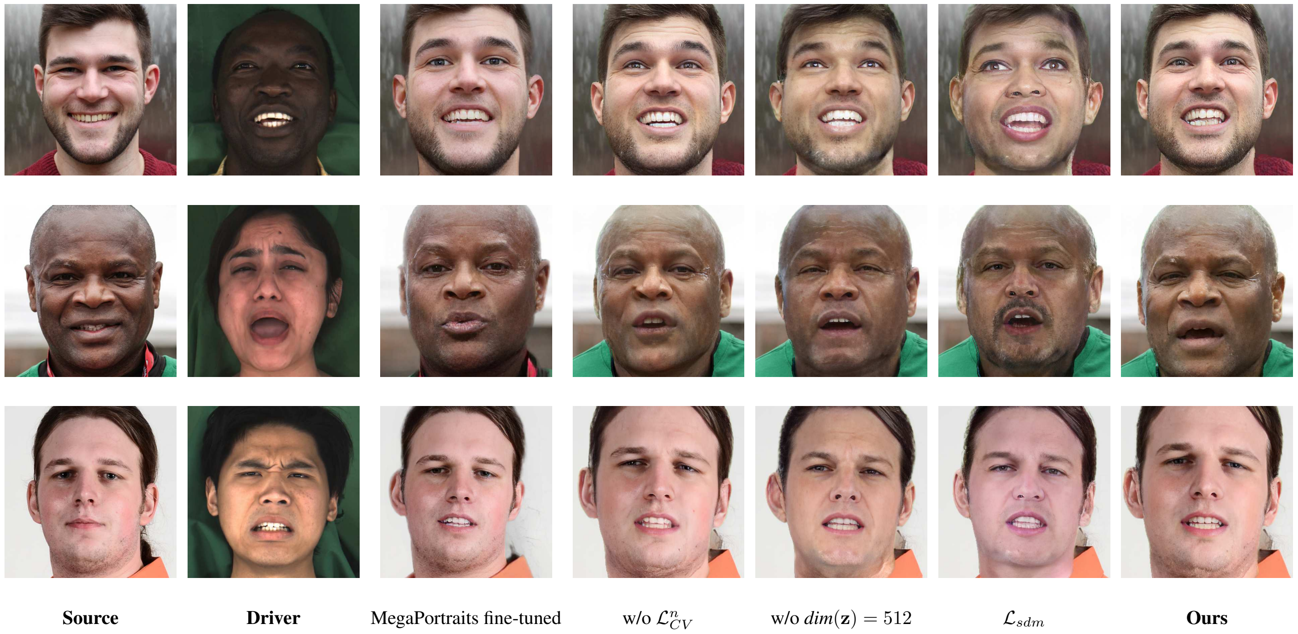

Figure 7. Visual comparison for our ablation study Fig. 7.

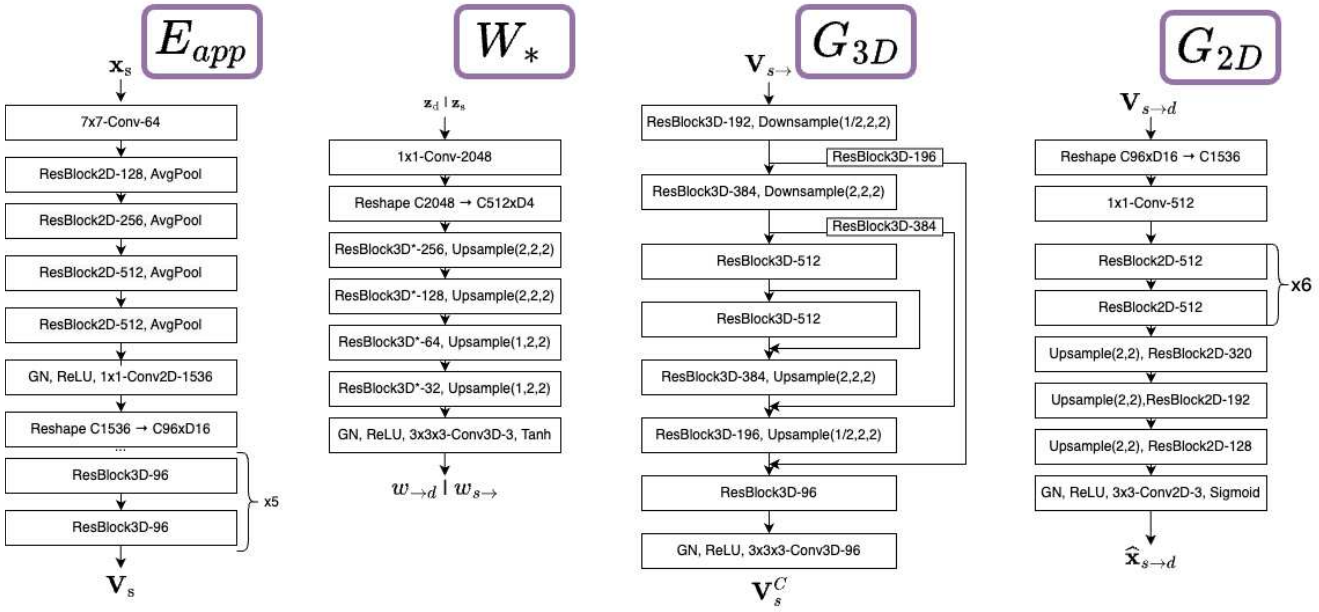
E_{app}
$\mathbf{x}_s$
7x7-Conv-64
ResBlock2D-128, AvgPool
ResBlock2D-256, AvgPool
ResBlock2D-512, AvgPool
ResBlock2D-512, AvgPool
GN, ReLU, 1x1-Conv2D-1536
Reshape C1536 → C96xD16
...
ResBlock3D-96
ResBlock3D-96
x5
$\mathbf{V}_s$
W_*
$\mathbf{z}_d \mid \mathbf{z}_s$
1x1-Conv-2048
Reshape C2048 → C512xD4
ResBlock3D*-256, Upsample(2,2,2)
ResBlock3D*-128, Upsample(2,2,2)
ResBlock3D*-64, Upsample(1,2,2)
ResBlock3D*-32, Upsample(1,2,2)
GN, ReLU, 3x3x3-Conv3D-3, Tanh
$w_{\to d} \mid w_{s\to}$
G_{3D}
$\mathbf{V}_{s\to}$
ResBlock3D-192, Downsample(1/2,2,2)
ResBlock3D-196
ResBlock3D-384, Downsample(2,2,2)
ResBlock3D-384
ResBlock3D-512
ResBlock3D-512
ResBlock3D-384, Upsample(2,2,2)
ResBlock3D-196, Upsample(1/2,2,2)
ResBlock3D-96
GN, ReLU, 3x3x3-Conv3D-96
$\mathbf{V}_s^C$
G_{2D}
$\mathbf{V}_{s\to d}$
Reshape C96xD16 → C1536
1x1-Conv-512
ResBlock2D-512
ResBlock2D-512
x6
Upsample(2,2), ResBlock2D-320
Upsample(2,2),ResBlock2D-192
Upsample(2,2), ResBlock2D-128
GN, ReLU, 3x3-Conv2D-3, Sigmoid
$\hat{\mathbf{x}}_{s\to d}$

Figure 8. Architectures of main components of our main model

References

[1] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. Voxceleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 1, 3

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[3] Nikita Drobyshev, Jenya Chelishev, Taras Khakhulin, Aleksei Ivakhnenko, Victor Lempitsky, and Egor Zakharov. Megaportraits: One-shot megapixel neural head avatars, 2023. 1, 2, 5

[4] Tobias Fischer, Hyung Jin Chang, and Y. Demiris. Rt-gene: Real-time eye gaze estimation in natural environments. In *ECCV*, 2018. 1

[5] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent

Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017. 4

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. 4

[7] Xueting Li, Shalini De Mello, Sifei Liu, Koki Nagano, Umar Iqbal, and Jan Kautz. Generalizable one-shot neural head avatar, 2023. 5

[8] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 2, 3

[9] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, 2015. 1, 3

[10] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. 3

[11] Aliaksandr Siarohin, Stéphane Lathuilière, S. Tulyakov, Elisa Ricci, and N. Sebe. First order motion model for image animation. *ArXiv*, abs/2003.00196, 2019. 5

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015. 1

[13] H. Wang, Yitong Wang, Zheng Zhou, Xing Ji, Zhifeng Li, Dihong Gong, Jin Zhou, and Wenyu Liu. Cosface: Large margin cosine loss for deep face recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5265–5274, 2018. 2

[14] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[15] Fei Yin, Yong Zhang, Xiaodong Cun, Mingdeng Cao, Yanbo Fan, Xuan Wang, Qingyan Bai, Baoyuan Wu, Jue Wang, and Yujiu Yang. Styleheat: One-shot high-resolution editable talking face generation via pre-trained stylegan, 2022. 5

[16] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation, 2018. 2, 3

[17] Wangbo Yu, Yanbo Fan, Yong Zhang, Xuan Wang, Fei Yin, Yunpeng Bai, Yan-Pei Cao, Ying Shan, Yang Wu, Zhongqian Sun, and Baoyuan Wu. Nofa: Nerf-based one-shot facial avatar reconstruction, 2023. 2, 5

[18] Bowen Zhang, Chenyang Qi, Pan Zhang, Bo Zhang, Hsiang-Tao Wu, Dong Chen, Qifeng Chen, Yong Wang, and Fang Wen. Metaportrait: Identity-preserving talking head generation with fast personalized adaptation, 2023. 5

[19] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5738–5746, 2019. 4

[20] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017. 2