SadTalker: Learning Realistic 3D Motion Coefficients for Stylized Audio-Driven Single Image Talking Face Animation
Supplemental Material

Wenxuan Zhang*, [1, 2] Xiaodong Cun*, [3] Xuan Wang[4] Yong Zhang[3] Xi Shen[3]
Yu Guo[2, 5] Ying Shan[3] Fei Wang†, [2, 5]
[1] School of Software Engineering, Xi'an Jiaotong University
[2] National Key Laboratory of Human-Machine Hybrid Augmented Intelligence
[3] Tencent AI Lab [4] Ant Group [5] IAIR, Xi'an Jiaotong University

A. Additional Experiments

A.1. PIRenderer *v.s* Our FaceRender on Face Reenactment

We compare our FaceRender and PIRenderer [8] on the task of video-driven face reenactment. We have already shown the visual comparisons in Fig 9 of the main paper, here, we give the numerical comparison results on the HDTF dataset. We evaluate these two methods using cross-identity settings and the results are conducted over 354 videos. As shown in Tab A1, the proposed method shows a much better visual quality in terms of FID and CSIM, which demonstrates the advantage of the proposed methods for audio-driven talking-head generation. For more differences between the proposed method and PIRenderer for this task, we also discuss the influence of the face alignment coefficient in Sec. B.4.

Method	FID ↓	CPBD ↑	CSIM ↑
PIRenderer [8]	26.521	**0.363**	0.857
Our face render	**19.646**	0.334	**0.880**

Table A1. Face render evaluation.

A.2. Cross-ID Settings and More Test Datasets

We conduct the *same-identity* experiment in the main paper, where the first frame of the test video is regarded as the reference image and the corresponding audio is regarded as the driving signal, generating a video that has the synchronized expression but diverse head poses. Differently, in the cross-identity experiment, the driving audio comes from another video. This kind of setting is also widely used in the comparison of the video-driven face reenactment [9]. In the cross-identity experiment, the reference pose of PC-AVS comes from the audio's corresponding video.

To this end, besides the HDTF [13] dataset in the main paper (we evaluate the results under the same-identity experiment), we also evaluate our method on HDTF and VoxCeleb2 [6] datasets in ***cross-identity setting*** as in Table A2 and Table A3. VoxCeleb2 contains over 1 million utterances of 6112 speakers, in which there are 36k utterances of 118 speakers in the test set. We randomly select 3 videos for each speaker, obtaining 354 videos as for evaluation. The evaluation metrics are the same as those the same-identity experiment on the HDTF dataset. We directly evaluate the pretrained model of all the models on this dataset. We crop the videos in the same way used in [9] and resize the frames to 256×256. As shown in the Tables, the proposed method shows better lip synchronization in this kind of setting on both datasets in most metrics. The same trend is also observed in the head motion and visual quality of the final videos.

*Equal contribution
†Corresponding Author

Method	Lip Synchronization		Learned Head Motion		Video Quality		
	LSE-C↑	LSE-D↓	Diversity↑	Beat Align↑	FID↓	CPBD↑	CSIM↑
Real Video	8.211	6.982	0.259	0.271	0	0.428	1.000
Wav2Lip* [7]	9.641	6.035	N./A.	N./A.	21.727	0.368	0.846
PC-AVS** [14]	8.959	6.435	N./A.	N./A.	99.098	0.201	0.648
MakeItTalk [15]	4.937	10.231	0.2553	0.276	26.829	0.333	0.834
Audio2Head [10]	7.237	**7.648**	0.1783	0.260	24.404	0.282	0.818
Wang *et al.* [11]	4.634	10.457	0.2260	0.265	22.302	0.294	0.805
Ours	**7.343**	7.709	**0.2759**	**0.284**	**20.886**	**0.334**	**0.846**

Table A2. Comparison with the state-of-the-art method on HDTF dataset [13] with *cross-identity* setting. Wav2Lip* achieves the best video quality since it only animates the lip region while other regions are the same as the original frame. In cross-identity setting, PC-AVS** is evaluated using the reference pose from the driving video and fails in some samples.

Method	Lip Synchronization		Learned Head Motion		Video Quality		
	LSE-C↑	LSE-D↓	Diversity↑	Beat Align↑	FID↓	CPBD↑	CSIM↑
Real Video	6.209	7.911	0.4879	0.266	0	0.099	1.000
Wav2Lip* [7]	7.640	7.099	N./A.	N./A.	19.293	0.107	0.936
PC-AVS** [14]	7.168	7.443	N./A.	N./A.	111.043	0.074	0.494
MakeItTalk [15]	3.756	10.222	**0.5230**	0.275	23.501	0.063	0.883
Audio2Head [10]	5.266	8.788	0.2064	0.273	54.694	0.098	0.602
Wang *et al.* [11]	3.441	10.519	0.2547	0.272	42.092	**0.136**	0.750
Ours	**5.571**	**8.503**	0.5211	**0.277**	**22.738**	0.081	**0.893**

Table A3. Comparison with the state-of-the-art method on VoxCeleb2 [6] dataset under *cross-identity* setting. Wav2Lip* achieves the best video quality since it only animates the lip region while other regions are the same as the original frame. In cross-identity setting, PC-AVS** is evaluated using the reference pose from the driving video and fails in some samples.

A.3. Emotion Control

In ExpNet, although we mainly focus on lip movement and eye blinking, our method can also control the emotion of the generated video without re-design. As shown in Fig. A1, the emotion of generated video keeps the same with source image. Thus, we can perform emotion transfer through the source image, and then a talking-head video with different emotions can be produced in our system.

B. More Implementation Details

We provide the detailed dataset pre-processing, network structures, loss functions, the discussion on alignment coefficients and how to generate long sequence using PoseVAE in Sec. B.1, Sec. B.2, Sec. B.3, Sec. B.4 and Sec. B.5.

B.1. Dataset Pre-processing Details

Audio We follow Wav2Lip [7] to pre-process the audio. Specifically, we pre-process all the audio to 16k Hz. Then, we convert it to the mel-spectrograms with FFT window size 800, hop length 200 and 80 Mel filter banks. Thus, for each frame, we have 0.2s mel-spectrogram feature with the shape of 16×80.

Lip-only coefficients To avoid the disturbance of other facial expression (*e.g.*, emotions), we regard lip-only expression coefficients as the ground truth to train ExpNet. Since 3DMM expression coefficients are mixed, we first perform Wav2Lip [7] on *a single image* and the audio, which generates the still video with only lip movement, and then, extract the *full facial expression coefficients* on the generated video. These expression coefficients represent lip-only animation with relatively still upper faces.

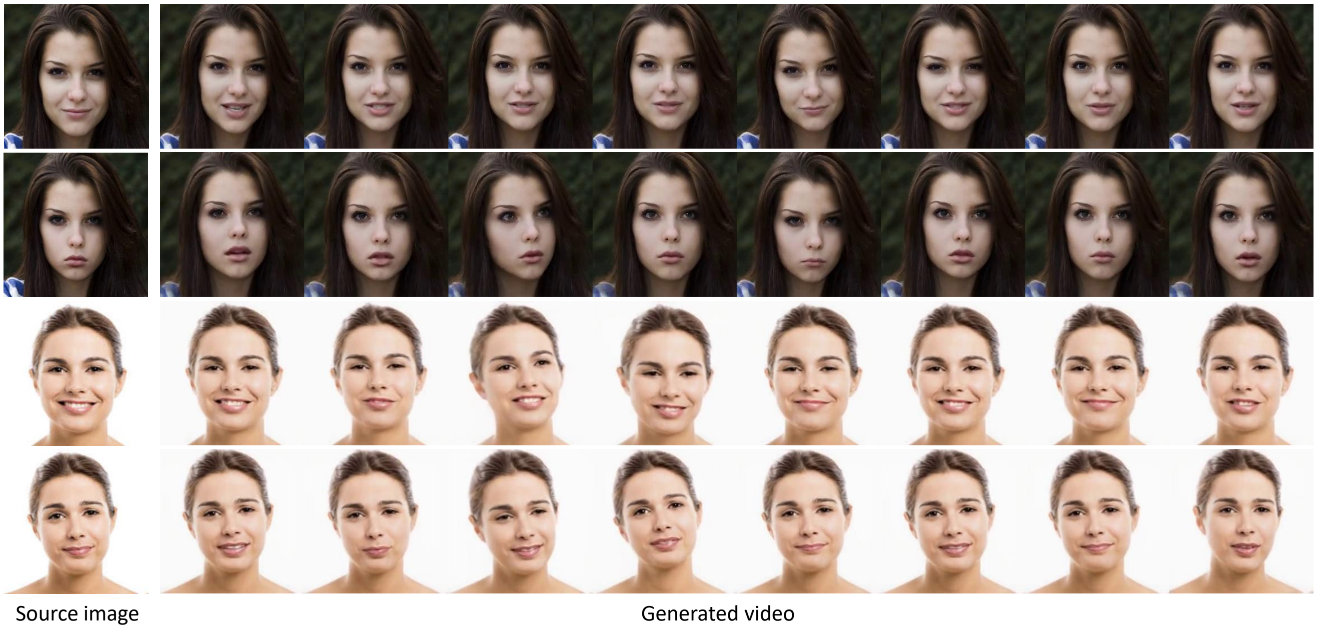

Figure A1. The generated video can retain the emotion of the source image more effectively.

B.2. Network Structure Details

ExpNet Our ExpNet is built via an audio encoder Φ_A and a linear layer Φ_M. We use the parameters from the pre-trained Wav2Lip to initialize the audio encoder. As discussed in the main paper, we first encode the audio feature into an audio embedding. And then, we generate the expression coefficients $\beta^g_{\{1,...,t\}}$ with additional coefficients of the first frame β_0 and blink control signal z_{blink}. As shown in Fig. B2 (c), the audio encoder Φ_A is built via a four stages ResBlock-C as in Fig. B2 (a). we only use a single linear layer Φ_M as in Fig. B2 (b).

PoseVAE As shown in Fig. B2 (e), both the encoder and decoder of our PoseVAE contain several linear layers. For the conditions, the encoder μ and $\sum$ is mapped through the concatenation of the $\Delta\rho_{\{1:T\}}$, the 46 dimensional one-hot vector Z_{style} (our training dataset contains 46 identities) and the encoded features from audio encoder Φ_A. As for the decoder, we first add the re-parameterized feature and the style embedding. Then, we concatenate the audio feature similar to the encoder.

FaceRender As discussed in the main paper of Fig. 5, our face render is inspired by the motion transfer method face-vid2vid [12]. We introduce a mappingNet to remap the learned 3DMM motion coefficients to the space of unsupervised 3D keypoints. As shown in Figure B2 (d), the mapping network contains the t-frames ($[t-2:t+2]$) motion coefficients of pose $\rho_{[t-2:t+2]}$ and expression $\beta_{[t-2:t+2]}$ to generate the motions representation of face-vid2vid [12] (`yaw`, `pitch`, `roll`, `tr`, and δ) in frame t. Other networks in our FaceRender have the same structures in [12]. Please refer to face-vid2vid [12] for more network details about the FaceRender.

B.3. Loss Function Details

ExpNet As described in the main paper, we use the expression coefficients which are generated from the pre-trained wav2lip [7] and then perform 3D face capture [1] as guidance (lip-only expression coefficients for short). Basically, we calculate $\mathcal{L}_{distill}$ through the Mean-Squared loss between lip-only expression coefficients and the generated expression coefficients in training. Formally, a T-frames $\mathcal{L}_{distill}$ can be written as:

$$\mathcal{L}_{distill} = \frac{1}{T}\sum_{t=1}^{T}\left(\beta^g_t - \beta^{lip}_t\right)^2 \tag{1}$$

Where β^{lip}_t and β^g_t are the lip only and the generated expression coefficients, respectively.

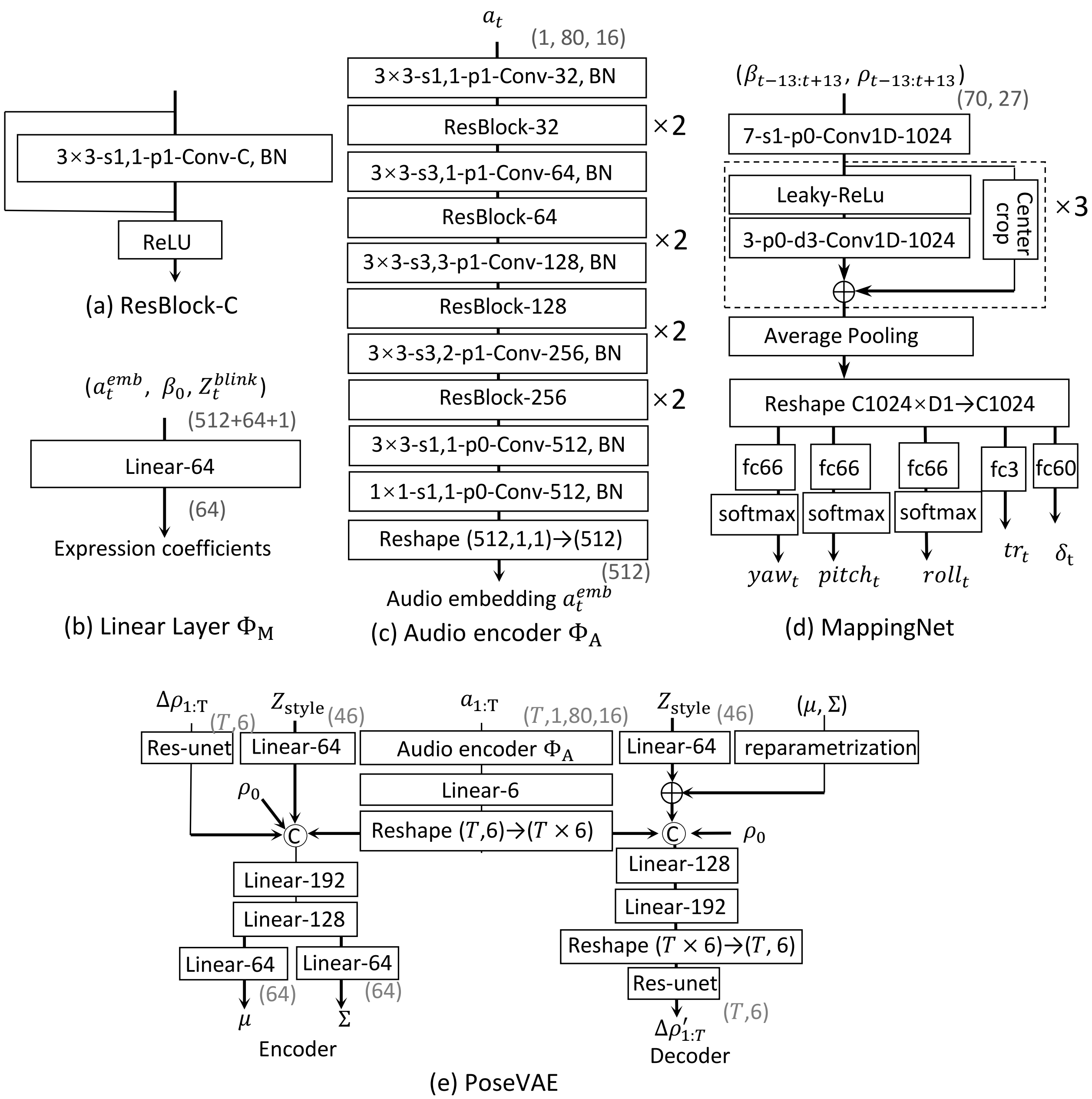

Figure B2. The architectures of the networks in our model. Here, '3×3-s1,1-p1-d1-Conv-32' means a convolutional layer with the kernel size 3×3, the stride size (1,1), padding size (1,1), the dilation size (1,1) and the output channel is 32.

We also calculate the loss function on the projected 2D landmarks of the rendered 3D face. In detail, as shown in Fig. B3, the height and width of the eye area in the t-th frame are defined as follows:

$$E_t^w = \frac{\left\|P_t^{39} - P_t^{36}\right\|_2 + \left\|P_t^{45} - P_t^{42}\right\|_2}{2} \tag{2}$$

$$E_t^h = \frac{\left\|P_t^{37} + P_t^{38} - P_t^{40} - P_t^{41}\right\|_2}{2} \tag{3}$$

$$+\frac{\left\|P_t^{43} + P_t^{44} - P_t^{46} - P_t^{47}\right\|_2}{2}. \tag{4}$$

Where E_t^w is the width of the eye area in frame t, E_t^h is the width of the eye area in frame t, P_t^i is the i-th landmark in frame t.

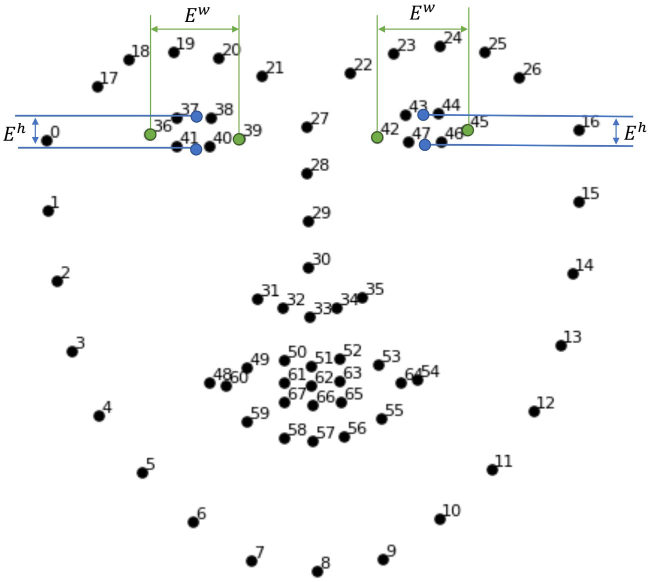

Figure B3. Face landmarks visualization.

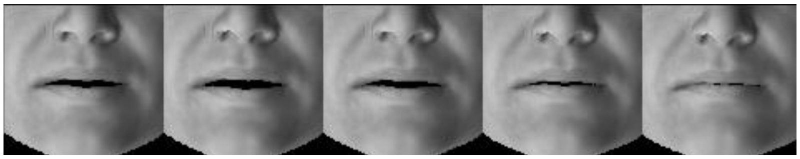

Figure B4. Example of the cropped interesting region in the 3D rendered face sequences to calculate the lip-reading loss.

we define $R_t = \frac{E_t^h}{E_t^w}$ as the predicted and calculate eye loss as follows:

$$\mathcal{L}_{eye} = \sum_{t=1}^{T} \left\| R_t - Z_t^{blink} \right\|_1, \tag{5}$$

where Z_t^{blink} is the eye blinking control signal the of t-th frame which is generated uniformly and randomly. To eliminate the effects of $\mathcal{L}_{eye}$ on other facial expression, we also constrain the minimal modification in the other landmarks. Thus,

$$\mathcal{L}_{lks} = \lambda_{eye}\mathcal{L}_{eye} + \frac{1}{T}\frac{1}{N}\sum_{t=1}^{T}\sum_{i=1}^{M} \left\| P_t^i - P_t^{i\prime} \right\|_2^2, \tag{6}$$

where λ_{eye} is set to 200, $P_t^{i\prime}$ is the landmarks predicted by the lip-only expression coefficients, $\mathcal{M}$ is a set of landmarks other than the eye areas.

Besides, we use pretrained lip-reading models proposed in [5] to calculate lip reading loss $\mathcal{L}_{read}$ inspired by [2]. We use the pretrained video-based lip-reading model where the input is a sequence (5 frames in our case) of the cropped interesting region around the mouth (as shown in Fig B4) and the target is a series of the character sequence. So we employ a differentiable 3D face render [1] in ExpNet to render the images through the generated expression coefficients, then, we crop the mouth area of the rendered images using the bounding box of the mouth landmarks, obtaining the logit of the character sequences $\mathbf{C_p}$. As for the supervision, we generate the logit of the character sequences $\mathbf{C_{gt}}$ from the ground truth audio using the audio-driven lip-reading model. Thus, our goal is to minimize the difference between $\mathbf{C_p}$ and $\mathbf{C_{gt}}$. In other words,

$$\mathcal{L}_{read} = \text{CrossEntrory}(\mathbf{C_{gt}}, \mathbf{C_p}) \tag{7}$$

Overall, the final loss of ExpNet is given by :

$$\mathcal{L}_{exp} = \lambda_{distill}\mathcal{L}_{distill} + \lambda_{read}\mathcal{L}_{read} + \lambda_{lks}\mathcal{L}_{lks} \tag{8}$$

Where $\lambda_{distill}$, λ_{read}, λ_{lks} are set to 2, 0.01, and 0.01, respectively.

PoseVAE We first calculate the reconstruction loss by applying Mean-Squared loss between the generated $\Delta\rho'_{\{1...T\}}$ and the original $\Delta\rho_{\{1...T\}}$:

$$\mathcal{L}_{MSE} = \frac{1}{T}\sum_{t=1}^{T}(\Delta\rho'_t - \Delta\rho_t)^2 \tag{9}$$

Meanwhile, we encourage the similarity of the latent space distribution and the Gaussian distribution with the mean vector μ and covariance matrix $\sum$. So we define $\mathcal{L}_{KL}$ as the Kullback–Leibler (KL) divergence between the latent space distribution and the Gaussian distribution. We also employ a discriminator D based on the PatchGAN [4] to perform 1D convolution on the head motion sequence as Speech2Gesture [3]. We define the adversarial loss $\mathcal{L}_{GAN}$:

$$\mathcal{L}_{GAN} = \arg\min_{G}\max_{D}(G, D) \tag{10}$$

Where G is proposed PoseVAE. The total loss of PoseVAE can be summarized as follows.

$$\mathcal{L}_{pose} = \lambda_{MSE}\mathcal{L}_{MSE} + \lambda_{KL}\mathcal{L}_{KL} + \lambda_{GAN}\mathcal{L}_{GAN} \tag{11}$$

where λ_{MSE}, λ_{KL} and $\mathcal{L}_{GAN}$ are set to 1, 1, and 0.7, respectively.

FaceRender We add a MappingNet to map the explicit 3DMM coefficients to the space of the face-vid2vid [12], to training, apart from the loss functions used in face-vi2vid [12], we add L_1 regularization on the domain of unsupervised keypoints:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{n=1}^{N}||K'_n - K_n||_1, \tag{12}$$

where K'_n and K_n are the n-th keypoint generated by our MappingNet and the motion generator of the original face-vid2vid, respectively. The weight of $\mathcal{L}_1$ is set to 20, and the weights of the other loss functions keep the same as in face-vid2vid and they are calculated on the final generated image. Please refer to face-vid2vid [12] for more details.

B.4. More Details about the Alignment Coefficients

In the main paper, we show the effect of the alignment coefficients in Fig 9. Here, we give more details about the alignment coefficients. Generally, the alignment coefficients are the transformation parameters (translation and scaling) to transform and crop the arbitrary video to the aligned face video for deep 3D face reconstruction [1]. The implicit modulation of PIRenderer [8] contains 73 dimensional motion coefficients, including the expression (64), head pose (6) and the alignment coefficients (3). Since their method focuses on video driving animation, the alignment coefficients are known in testing. However, it is hard to *learn* from the audio since there is no relationship between the alignment and the audio. We also try to learn and use the alignment coefficients of the first frame in our method (as shown in Fig. 9 and the supp. video), however, the produced head motion is aligned and unnatural. We discard it to obtain more natural video results. Thus, our motion coefficients (70) only contains the expression (64) and head pose (6).

B.5. Generating Long sequence in PoseVAE

We predict the residual of the 32 continuous head pose coefficients $[\Delta\rho_1, .., \Delta\rho_{32}]$ from 32-frame audio features $[a_1, .., a_{32}]$, the reference head pose ρ_0 and the style code Z_{style}, generating the head pose coeffs $[\rho_1, .., \rho_{32}]$ = $[\Delta\rho_1, .., \Delta\rho_{32}] + \rho_0$. To generate longer sequence, *e.g.*, $[\rho_{33}, ..., \rho_{64}]$, we still use ρ_0 as the reference to guarantee the stability of generated pose sequence.

C. Supplementary Video

We provide a supplementary video to include all the video results of our method and other related methods as comparisons, the ablation study of each component, and more results of our method in different languages. Please refer to https://sadtalker.github.io for the supplementary video.

References

[1] 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 *CVPR Workshops*, 2019. 3, 5, 6

[2] Panagiotis P. Filntisis, George Retsinas, Foivos Paraperas-Papantoniou, Athanasios Katsamanis, Anastasios Roussos, and Petros Maragos. Visual speech-aware perceptual 3d facial expression reconstruction from videos. *arXiv preprint arXiv:2207.11094*, 2022. 5

[3] Shiry Ginosar, Amir Bar, Gefen Kohavi, Caroline Chan, Andrew Owens, and Jitendra Malik. Learning individual styles of conversational gesture. In *CVPR*, 2019. 6

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017. 6

[5] Pingchuan Ma, Yujiang Wang, Stavros Petridis, Jie Shen, and Maja Pantic. Training strategies for improved lip-reading. In *ICASSP*, 2022. 5

[6] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: a large-scale speaker identification dataset. In *INTERSPEECH*, 2017. 1, 2

[7] K R Prajwal, Rudrabha Mukhopadhyay, Vinay P.Namboodiri, and C.V.Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *ACM MM*, 2020. 2, 3

[8] Yurui Ren, Ge Li, Yuanqi Chen, Thomas H Li, and Shan Liu. Pirenderer: Controllable portrait image generation via semantic neural rendering. In *ICCV*, 2021. 1, 6

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

[10] Suzhen Wang, Lincheng Li, Yu Ding, Changjie Fan, and Xin Yu. Audio2head: Audio-driven one-shot talking-head generation with natural head motion. In *IJCAI*, 2021. 2

[11] Suzhen Wang, Lincheng Li, Yu Ding, and Xin Yu. One-shot talking face generation from single-speaker audio-visual correlation learning. In *AAAI*, 2022. 2

[12] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. One-shot free-view neural talking-head synthesis for video conferencing. In *CVPR*, 2021. 3, 6

[13] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *CVPR*, 2021. 1, 2

[14] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *CVPR*, 2021. 2

[15] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. Makelttalk: speaker-aware talking-head animation. *ACM Transactions on Graphics (TOG)*, 2020. 2