

AniTalker: Supplementary Materials

Anonymous Authors

1 DATA PROCESSING PIPELINE

Our data collection pipeline contains in four distinct stages, utilizing the datasets VoxCeleb [9], HDTF [24], and VFHQ [21]:

- (1) **Re-downloading Original Datasets:** To ensure uniform processing, given the different data handling methods across datasets, we downloaded the original videos. For VoxCeleb and HDTF, changes in the original sources meant we could only secure about 60-70% of the initial datasets. The VFHQ authors provided the complete set of videos, obviating the need for re-downloading.
- (2) **Face Detection:** This step involves detecting faces in videos. In contrast to previous studies, we chose not to align the faces to allow for positional shifts within the frame, aiming to preserve natural head movements.
- (3) **Applying Filtering Rules:** Our filtering process involved two main criteria. We first excluded faces with resolutions lower than 256×256 . Then, we conducted blur detection using the Laplacian operator and angle detection, excluding faces with a yaw angle greater than 60 degrees.
- (4) **Selecting Video Clips Based on Identity ID Tags:** To ensure a diversity of identities for the robust training of the identity encoder, we randomly selected 2-3 video clips per ID.
- (5) **Resize to 256×256 :** All our images, whether for training or testing, are originally based on the resolution of 256×256 . Therefore, the purpose of this step is to resize all images to 256×256 .

Finally, our efforts yielded a dataset containing 4,242 unique speaker IDs, encompassing a total of 17,108 video clips with a cumulative duration of 55 hours. Additionally, since the VFHQ dataset lacks an audio track, it was used exclusively during the motion representation training phase.

Table 1 provides comparative metrics of our dataset against those of EMO [16] and GAIA [5]. As outlined in the table, our dataset contains roughly a quarter of the unique human identifiers found in GAIA and about one-fifth of the training hours compared to EMO. Furthermore, our algorithm can be trained from scratch and does not rely on parameter initialization.

2 TRAINING DETAILS

2.1 Data Augmentation

For source and target images, no data augmentation strategies can be applied. The objective of this constraint is to ensure the consistency of the background, allowing the motion latent space to capture human motion without background movement. However, when training the identity encoder using metric learning, it becomes necessary to introduce some negative candidates. At this point, we can employ standard augmentation techniques, which

Table 1: Comparison of training dataset statistics. "#IDs": Number of unique human identifiers. "#Clips": Total number of video segments. "Hours": Total training hours. "TFS": Training from scratch. "-" indicates that this information is not provided.

Method	#IDs	#Clips	Hours	TFS
GAIA [5]	15,969	-	1,169	✓
EMO [16]	-	-	250	×
AniTalker	4,242	17,108	55	✓

include horizontal flipping, color jitter, Gaussian blur, shifting, scaling, and rotation. The specific implementations of these techniques are facilitated by the tools available at this URL ¹.

2.2 Training Configuration

2.2.1 Training Loss. Metric Learning Loss For the Triplet Loss [2], we set the margin to 0.01 and use the L2 distance metric. For the angular additive margin softmax (AAMSoftmax [17]), we set the margin m to 0.2 and the scaling factor s to 30, utilizing cosine distance.

Mutual Information Decoupling Loss We use the Contrastive Log-ratio Upper Bound (CLUB) [1] for mutual information (MI) minimization in high-dimensional spaces where only samples are available, not distribution forms. CLUB uses contrastive learning to estimate MI by contrasting conditional probabilities between positive and negative sample pairs. A variational form of CLUB (vCLUB) is developed for scenarios where the conditional distribution $p(y|x)$ is unknown, using a neural network to approximate $p(y|x)$. CLUB is further accelerated using a negative sampling strategy, enhancing computational efficiency while maintaining reliable MI estimation capabilities. Details can be found in this repo ². Here, we use the CLUB estimator to measure the differences between identity and motion distributions and minimize them.

Loss for Training Motion Representation This is also our primary loss function for training the motion encoder during the first phase. We utilize various types of losses, mainly comprising losses related to reconstruction (reconstruction loss L_{recon} , perceptual loss L_{percep}), adversarial loss (L_{adv}), mutual information loss (L_{MI}), and identity metric learning loss (L_{ML}). The specific forms of the reconstruction, perceptual, and adversarial losses are consistent with those used in LIA [20]. The overall loss is a weighted sum of these individual losses, as shown below:

$$L_{motion} = L_{recon} + \lambda_1 L_{percep} + \lambda_2 L_{adv} + \lambda_3 L_{MI} + \lambda_4 L_{ML}$$

where the values of λ_1 , λ_2 , λ_3 , and λ_4 are 0.1, 1, 0.1, and 0.1 respectively, in our experiment.

¹<https://github.com/albumentations-team/albumentations>

²<https://github.com/Linear95/CLUB/>

Loss for Training Motion Generator In our model, the generation loss

$$L_{\text{gen}} = L_{\text{diff}} + \lambda \sum_{k=1}^K L_{\text{var}_k}$$

is structured around two sets of attributes (i.e., $K = 2$). The first set pertains to camera parameters, including the position of the face within the frame and the scale of the face. The second set is related to pose attributes, consistent with methods [3, 5]. Specifically, the L_{var_k} losses are L2 losses between the predicted values and the ground truth values. Details regarding the feature extraction and representation for camera parameters and pose will be discussed in the upcoming sections. Additionally, we set λ to 1 in this formulation.

For the training of diffusion models, we employed the simplified loss objective described in [6] for the training of DDPMs. During the training phase, we used 1000 timesteps, while in the inference stage, we utilized DDIM [14] acceleration with 50 timesteps. We did not employ additional performance-enhancing methods such as class-free guidance (CFG).

2.2.2 Training and Inference Hardware. For training, we utilized four A100 (40G) GPUs, training each phase until the loss converged. Besides computing the perceptual loss, we did not incorporate any pre-trained parameters. The first phase, focusing on motion representation training, converged relatively quickly, requiring approximately 50 hours. The second phase, where we employed an exponential moving average (EMA) to stabilize training, converged more slowly, taking about 120 hours. For inference, we utilized a GeForce RTX 3060 Ti (8G) GPU. The process begins by generating a motion sequence from audio, followed by frame-by-frame rendering, which can support up to several minutes of inference without triggering memory overflow errors. Specifically, the GPU with 8G VRAM can generate up to 3 minutes of video in one inference.

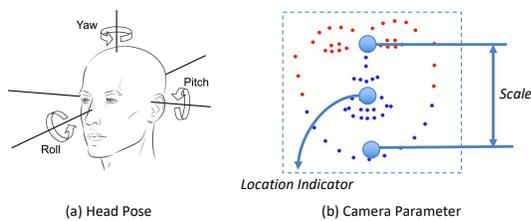


Figure 1: Controllable Attribute Features: Head Pose and Camera Parameter

2.2.3 Controllable Attribute Features. We utilize two types of features for controlling talking face generation. The first is the head pose, which includes yaw, pitch, and roll information, utilizing these three degrees of freedom to control the head's orientation, consistent with approaches found in [5, 11]. We use a pretrained extraction network³ to derive these three-dimensional features. Additionally, to further capture the facial variations within the frame, we considered two parameters: the face's position in the frame and its size, which indicates the distance from the camera. These

³https://github.com/cleardusk/3DDFA_V2

parameters are defined as camera parameters. Specifically, for the face's position, we use the x-coordinate of the nose landmark, as we observed that the face mostly moves horizontally rather than vertically. For the face's scale, we measure the distance from the eyebrows to the chin. These two attributes form a two-dimensional camera parameter feature. The visualization is illustrated in Figure 1.

Overall, during the training phase, we utilize several pre-trained models as attribute extractors for three attributes: head pose, head location, and head scale. In actual inference, since these features are explicit and interpretable, we can directly input specific values to control aspects such as the pitch angle, which can range from -90 to 90 degrees, to simulate head movements like nodding or looking up.

3 MODEL DETAILS

3.1 Identity and Motion Encoder

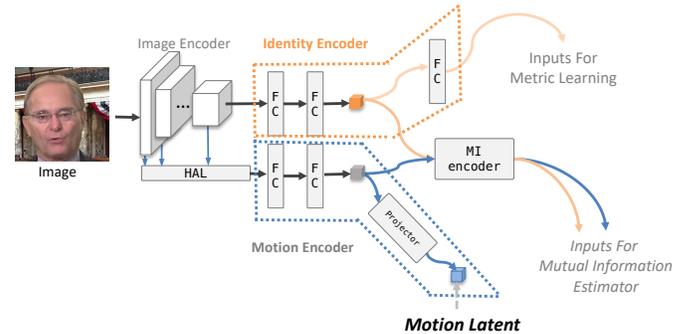


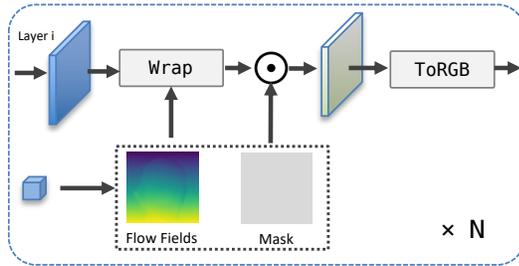
Figure 2: Detailed structure of the motion latent extraction

This section details the structures of the motion and identity encoders used during the first phase of training motion representations, as illustrated in Figure 2. The identity encoder is designed to learn identity information, with its outputs directed through a series of fully connected layers to a metric learning loss. Conversely, the motion encoder focuses on learning motion encoding, with its outputs routed to a loss function that calculates mutual information. This output later serves as an input for rendering and for generating results in the second phase. To reduce the dimensionality of the motion encoding and ensure a more compact representation, an additional projector is used, as indicated in the diagram. We tested two types of projectors: (1) **Direct Reduction**, which uses an FC layer to reduce the dimensionality of the hidden layer from 512 to a target dimension of 20, as depicted in the figure. (2) **Linear Motion Decomposition** [20], a method that represents motion in latent space by learning a set of orthogonal bases, each vector representing a fundamental visual transformation. The term "20 dimensions" refers to the number of distinct axes or directions in this space, which can be manipulated to encode different types of motion or transformations. In our experiments, we opted for the LMD approach. Comparative analysis and further discussion are presented in Section 3.4.2.

Table 2: Model Configuration on Speech Encoder (SE) and Diffusion Motion Generator (DMG)

Config.	SE	DMG
attention_dim	512	1024
attention_heads	2	2
# layers	4	2
dropout_rate	20%	20%
# params (M)	13	25

3.2 Rendering Block

**Figure 3: Rendering Block**

In our Rendering Block, we utilize the design of the G block from LIA [20], a wrap-based rendering module responsible for handling motion and features from different layers of the image encoder (derived from a portrait). This module samples the features, as depicted in the simplified structure in Figure 3. Here, the motion features generate a flow field that performs a wrap operation, creating deformed features based on features from layer i . These deformed features are then combined with a mask through dot product operations to restrict specific regions, followed by a toRGB conversion to produce the image. This module is executed N times, where N corresponds to the number of layers in the image encoder. In our experiments, we used $N=8$, with the width and height of the feature map from layer 1 to layer 8 being 256×256 , 128×128 , 64×64 , 32×32 , 16×16 , 8×8 , 4×4 , and 1×1 respectively.

3.3 Motion Generator

The motion generator aims to convert an audio input and a portrait into a motion sequence, which is then rendered by the image decoder. The main trainable modules involved are the speech encoder, variance adapter, and diffusion motion generator, which together create the motion latent. Both the speech encoder and the diffusion motion generator use the Conformer [4]⁴, with configuration details presented in Table 2.

The speech encoder processes downsampled audio features with dimensions $T \times C_1$, where C_1 is 512. Additionally, the input to the Diffusion Motion Generator consists of multiple components, formed by concatenating outputs from the speech encoder, the start motion latent and feature of the portrait, noisy latent, and time embedding. The start motion latent and feature of the portrait refer

⁴https://github.com/espnet/espnet/blob/master/espnet2/asr/encoder/conformer_encoder.py

Table 3: Comparison on the Capabilities of the Human Face Renderer

Method	Renderer	# params (M)	PSNR \uparrow	SSIM \uparrow
GAIA [5]	VAE ⁵	50	30.497	0.924
EMO [16]	VAE ⁶	84	33.114	0.960
AniTalker	Wrap-based	50	35.634	0.979

Table 4: Parameter search on the projector of the motion encoder

Method	FC	LMD	Dim.	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	CSIM \uparrow
AniTalker	\checkmark		32	28.653	0.899	0.081	0.924
		\checkmark	32	29.204	0.905	0.079	0.927
	\checkmark		20	28.387	0.895	0.083	0.922
		\checkmark	20	<u>29.071</u>	<u>0.905</u>	<u>0.079</u>	<u>0.927</u>
	\checkmark		10	27.685	0.879	0.089	0.914
		\checkmark	10	27.999	0.892	0.086	0.920

to the results of the portrait image encoder and the motion latent, respectively, with dimensions $T \times C_2$ and $T \times C_3$, where C_2 and C_3 are 512 and 20 in our experiments. The noisy latent is the noise-augmented motion latent with dimensions $T \times C_3$, and the time embedding is the diffusion time condition with dimensions $T \times 1$. These dimensions, except for the speech encoder, are unified to $T \times 128$ when input into the model. Ultimately, the Diffusion Motion Generator’s input includes a 1024-dimensional vector comprising the speech encoder’s input (512 dimensions), the start motion latent (128 dimensions), the start feature (128 dimensions), noisy latent (128 dimensions), and time embedding (128 dimensions).

The model parameters for the speech encoder, variance adapter, and diffusion motion generator are 13M, 2M, and 25M respectively, totaling 40M parameters for the motion generator stage.

3.4 Experiments

3.4.1 Analysis on Image Renderer. To compare rendering capabilities across different methods, we conducted a comparison with GAIA [5] and EMO [16], both employing a VAE [7]-based renderer. We randomly selected 100 images from the Celeb-A [8] face dataset to test reconstruction capabilities on human faces by extracting latent representations and then reconstructing based on these representations. For GAIA, since it is not open-sourced and to ensure a fair comparison, we used a structure similar to the one described in their paper and trained it with a model parameter size and dataset identical to our rendering module. Results in the table show that our method outperforms GAIA in facial reconstruction. Additionally, we compared our renderer with EMO’s, which adopts the architecture of Stable Diffusion [12]. Despite its larger parameter size, our results also demonstrate improvement in reconstruction. We attribute these outcomes to two main factors: firstly, the VAE-based perceptual compression [12] process tends to lose information, particularly high-frequency details such as hair

⁵Our reproduced result

⁶<https://huggingface.co/runwayml/stable-diffusion-v1-5>

strands. Secondly, VAEs are generally designed for generic static image generation tasks in Stable Diffusion and are not specifically tailored for representing human faces.

3.4.2 Analysis on the motion projector. To evaluate the structure of the motion projector and the impact of varying dimensions, we conducted a parameter search, as detailed in Table 4. First, to verify the effectiveness of Fully Connected (FC) and Linear Motion Decomposition (LMD) [20], we performed comparative experiments. From the table, LMD generally showed more favorable outcomes under any dimensions, which we attribute to its orthogonality. This orthogonality acts as a regularization method, implicitly enforcing separation between different features and thus enhancing performance.

Furthermore, to assess the impact of dimensionality on the model, we conducted experiments with three sets of dimensions: 10, 20, and 32. As the dimensions increased, the overall performance showed improvement. However, from 20 to 32, there was no significant enhancement, especially for metrics like SSIM, LPIPS, and CSIM, which showed no change. Therefore, all our experiments were based on a dimensionality of 20 combined with the Linear Motion Decomposition strategy.



Figure 4: Visual ablation study of identity and motion disentanglement using different methods.

3.4.3 Visual Ablation Study on Disentanglement. As a supplement to Section 4.4.1 of the main paper, we have randomly visualized several sets of disentanglement results, as illustrated in Figure 4. The objective is to drive the source portrait using the motion of the target portrait. In the absence of any metric learning or mutual information loss constraints, the baseline results exhibit issues of identity leakage, as shown in the third column of the figure. Implementing Euclidean distance metric learning or angle-based metric learning can mitigate the leakage to some extent, but problems still persist, as depicted in columns four and five. Furthermore, by incorporating mutual information loss on top of angle-based metric learning, the leakage issues are significantly alleviated, as demonstrated in the last column of the figure.

4 DEMO SETUP

To further illustrate the effectiveness of our experiments, we have prepared an extensive set of demonstrations available at AniTalker Project Page ⁷. Below, we provide a detailed explanation of the demo page setup:

- (1) **Audio-driven Talking Face Generation (Realism):** The input consists of audio plus random noise. The variance adapter does not receive any control signals, aiming to test the model's capability to generate realistic human faces.
- (2) **Audio-driven Talking Face Generation (Statue/Cartoon):** Similar to the realism setup, this demo tests the model with statues, reliefs, and cartoon characters. The results demonstrate our method's strong generalization capabilities.
- (3) **Video-driven Talking Face Generation (Cross/Self Reenactment):** To test the reconstruction effectiveness of motion representations, both identity-consistent and cross-identity tests are conducted. Motions from another/same person are used to drive a particular portrait without involving audio.
- (4) **Diversity:** To test the impact of diffusion noise on the results, we initially used two different random seeds, followed by nine different random seeds. The results demonstrate that while maintaining the consistency of the generated effects, noise can produce diverse outcomes.
- (5) **Controllability:** Testing the controllability of the variance adapter, we examined the results of combined control over pose, head location, and audio.
- (6) **Long Video Generation:** For long video generation, two cases are considered. We first generate text with ChatGPT ⁸, then use Text-to-speech (TTS) ⁹ to convert them to audio. The reading audio drives the portrait, testing the capability to generate long-duration videos. These videos, lasting several minutes, are generated on a GPU with only 8GB of VRAM (3060Ti), confirming that our algorithm does not rely on extensive computing resources.
- (7) **Method Comparison (Audio-driven):** Comparisons are made with baseline methods [18, 23, 25, 26] driven by audio.
- (8) **Method Comparison (Video-driven):** Comparisons are made with baseline methods [10, 13, 15, 19, 20, 22] driven by video, specifically comparing face reenactment techniques.
- (9) **Ablation Study:** To validate the impact of different modules on the outcomes, we tested four scenarios: mutual information decoupling, comparison with traditional representations, the variance adapter module, and the HAL module.

5 ETHICAL CONSIDERATION

The potential misuse of lifelike digital human face generation, such as for creating fraudulent identities or disseminating misinformation, necessitates preemptive ethical measures. Before utilizing these models, it is crucial for organizations to integrate ethical guidelines into their policies, ensuring the application of this technology emphasizes consent, transparency, and accountability. Furthermore, it is recommended to embed visible or invisible digital watermarks in any generated content.

⁷<https://anitalker.github.io/>

⁸<https://chat.openai.com/>

⁹<https://azure.microsoft.com/>

REFERENCES

- [1] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. Club: A contrastive log-ratio upper bound of mutual information. In *International Conference on Machine Learning (ICML)*. PMLR, 1779–1788.
- [2] Xingping Dong and Jianbing Shen. 2018. Triplet loss in siamese network for object tracking. In *Proceedings of the European conference on computer vision (ECCV)*, 459–474.
- [3] Chenpeng Du, Qi Chen, Tianyu He, Xu Tan, Xie Chen, Kai Yu, Sheng Zhao, and Jiang Bian. 2023. DAE-Talker: High Fidelity Speech-Driven Talking Face Generation with Diffusion Autoencoder. *Proceedings of the 31th ACM International Conference on Multimedia (ACM MM) (2023)*.
- [4] Anmol Gulati et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *Conference of the International Speech Communication Association (InterSpeech) (2020)*.
- [5] Tianyu He, Junliang Guo, Runyi Yu, Yuchi Wang, Jialiang Zhu, Kaikai An, Leyi Li, Xu Tan, Chunyu Wang, Han Hu, HsiangTao Wu, Sheng Zhao, and Jiang Bian. 2024. GAIA: Zero-shot Talking Avatar Generation.
- [6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems (2020)*.
- [7] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114 (2013)*.
- [8] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
- [9] Arsha Nagrani, Joon Son Chung, and Andrew Senior. 2017. Voxceleb: a large-scale speaker identification dataset. *arXiv preprint arXiv:1706.08612 (2017)*.
- [10] Youxin Pang, Yong Zhang, Weize Quan, Yanbo Fan, Xiaodong Cun, Ying Shan, and Dong-ming Yan. 2023. Dpe: Disentanglement of pose and expression for general video portrait editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 427–436.
- [11] Konpat Preechakul et al. 2022. Diffusion Autoencoders: Toward a Meaningful and Decodable Representation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.
- [13] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. 2019. First order motion model for image animation. *Advances in neural information processing systems 32 (2019)*.
- [14] Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations (ICLR)*.
- [15] Jiale Tao, Biao Wang, Tiezheng Ge, Yuning Jiang, Wen Li, and Lixin Duan. 2022. Motion Transformer for Unsupervised Image Animation. In *European Conference on Computer Vision*. Springer, 702–719.
- [16] Linrui Tian, Qi Wang, Bang Zhang, and Liefeng Bo. 2024. EMO: Emote Portrait Alive - Generating Expressive Portrait Videos with Audio2Video Diffusion Model under Weak Conditions. arXiv:2402.17485 [cs.CV]
- [17] Feng Wang, Jian Cheng, Weiyang Liu, and Haijun Liu. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters* 25, 7 (2018), 926–930.
- [18] Suzhen Wang, Lincheng Li, Yu Ding, Changjie Fan, and Xin Yu. 2021. Audio2head: Audio-driven one-shot talking-head generation with natural head motion. *International Joint Conference on Artificial Intelligence (IJCAI) (2021)*.
- [19] Ting-Chun Wang, Arun Mallya, and Ming-Yu Liu. 2021. One-shot free-view neural talking-head synthesis for video conferencing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10039–10049.
- [20] Yaohui Wang, Di Yang, Francois Bremond, and Antitza Dantcheva. 2022. Latent image animator: Learning to animate images via latent space navigation. *Proceedings of the International Conference on Learning Representations (2022)*.
- [21] Liangbin Xie, Xintao Wang, Honglun Zhang, Chao Dong, and Ying Shan. 2022. 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*. 657–666.
- [22] Bohan Zeng, Xuhui Liu, Sicheng Gao, Boyu Liu, Hong Li, Jianzhuang Liu, and Baochang Zhang. 2023. Face Animation with an Attribute-Guided Diffusion Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 628–637.
- [23] Wenxuan Zhang, Xiaodong Cun, Xuan Wang, Yong Zhang, Xi Shen, Yu Guo, Ying Shan, and Fei Wang. 2023. SadTalker: Learning Realistic 3D Motion Coefficients for Stylized Audio-Driven Single Image Talking Face Animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8652–8661.
- [24] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. 2021. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [25] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. 2021. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*.
- [26] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. 2020. Makeltalk: speaker-aware talking-head animation. *ACM Transactions On Graphics (TOG) (2020)*.

465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580