

StyleTalk++: A Unified Framework for Controlling the Speaking Styles of Talking Heads

Suzhen Wang , Yifeng Ma , Yu Ding , Member, IEEE, Zhipeng Hu , Changjie Fan , Tangjie Lv , Zhidong Deng , Member, IEEE, and Xin Yu 

Abstract—Individuals have unique facial expression and head pose styles that reflect their personalized speaking styles. Existing one-shot talking head methods cannot capture such personalized characteristics and therefore fail to produce diverse speaking styles in the final videos. To address this challenge, we propose a one-shot style-controllable talking face generation method that can obtain speaking styles from reference speaking videos and drive the one-shot portrait to speak with the reference speaking styles and another piece of audio. Our method aims to synthesize the style-controllable coefficients of a 3D Morphable Model (3DMM), including facial expressions and head movements, in a unified framework. Specifically, the proposed framework first leverages a style encoder to extract the desired speaking styles from the reference videos and transform them into style codes. Then, the framework uses a style-aware decoder to synthesize the coefficients of 3DMM from the audio input and style codes. During decoding, our framework adopts a two-branch architecture, which generates the stylized facial expression coefficients and stylized head movement coefficients, respectively. After obtaining the coefficients of 3DMM, an image renderer renders the expression coefficients into a specific person’s talking-head video. Extensive experiments demonstrate that our method generates visually authentic talking head videos with diverse speaking styles from only one portrait image and an audio clip.

Index Terms—Deep learning, facial animation, head pose generation, neural network, neural rendering, talking head generation.

Manuscript received 20 April 2023; revised 26 October 2023; accepted 13 January 2024. Date of publication 24 January 2024; date of current version 7 May 2024. This work was supported in part by the 2022 Hangzhou Key Science and Technology Innovation Program under Grant 2022AIZD0054 and in part by the Key Research and Development Program of Zhejiang Province under Grant 2022C01011. This research was partially supported in part by the ARC-Discovery under Grant DP220100800 and in part by ARC-DECRA under Grant DE230100477. This work was supported in part by the National Science Foundation of China (NSFC) under Grant 62176134, in part by the Institute Guo Qiang, Tsinghua University, under Grant 2019GQG0002, and in part by research and application on AI technologies for smart mobility funded by SAIC Motor. An earlier version of this paper was presented in IJCAI 2021 [DOI: 10.24963/ijcai.2021/152], AAAI 2022 [DOI: 10.1609/aaai.v36i3.20154], and AAAI 2023 [DOI: 10.1609/aaai.v37i2.25280]. Recommended for acceptance by J. Gall. (Suzhen Wang and Yifeng Ma contributed equally to this work.) (Corresponding author: Yu Ding.)

Suzhen Wang, Yu Ding, Zhipeng Hu, Changjie Fan, and Tangjie Lv are with the Fuxi AI Lab, NetEase, Hangzhou, Zhejiang 310052, China (e-mail: wang-suzhen@corp.netease.com; dingyu01@corp.netease.com; zphu@corp.netease.com; fanchangjie@corp.netease.com; hzltangjie@corp.netease.com).

Yifeng Ma and Zhidong Deng are with the Department of Computer Science and Technology, BNRist, THUAI, State Key Laboratory of Intelligent Technology and Systems, Tsinghua University, Beijing 100190, China (e-mail: mayf18@mails.tsinghua.edu.cn; michael@tsinghua.edu.cn).

Xin Yu is with the School of Computer Science, The University of Queensland, Brisbane QLD 4072, Australia (e-mail: xin.yu@uq.edu.au).

Digital Object Identifier 10.1109/TPAMI.2024.3357808

I. INTRODUCTION

AUDIO-DRIVEN photo-realistic talking head generation has drawn growing attention due to its broad applications in virtual human creation, visual dubbing, and short video creation. The past few years have witnessed tremendous progress in accurate lip synchronization [2], [4], head pose generation [1], [5] and high-fidelity video generation [6], [7]. However, existing *one-shot* based works pay less attention to modeling diverse speaking styles, thus failing to produce expressive talking head videos with various styles.

The speaking styles of individuals consist of both facial expression style and head pose style, which respectively represent the spatial and temporal co-activations of full facial expressions and head poses. In real-world scenarios, different individuals may speak the same utterance with significantly diverse personalized speaking styles. Due to such significant diversities, creating controllable talking heads that showcase specific styles remains a great challenge, particularly in one-shot settings. Previous works [8], [9] have denoted speaking style simply as discrete emotion classes, which is insufficient for representing flexible speaking styles. Even though recent methods [10], [11] can control upper facial expressions by incorporating an additional emotional source video, they only transfer upper facial motion characteristics at a static frame level, ignoring the temporal dynamics of speaking styles. Therefore, a universal spatio-temporal representation of speaking styles is highly desirable.

In this paper, we propose a new method called **StyleTalk++** that can learn a comprehensive representation of speaking style from a talking video. Our approach aims to create stylized and realistic talking videos for a one-shot speaker image, where the speaker delivers the specified audio content with the extracted speaking style from style reference videos. To achieve this, our method utilizes a unified, style-controllable framework that first extracts the speaking style from a reference video and then embeds it into the audio-driven generated coefficients of a 3D Morphable Model (3DMM). These coefficients include facial expression and head pose parameters, and as a result, the unified framework is instantiated into two branches for stylized facial expression generation and head pose generation, respectively. Finally, an image renderer takes facial animations, head poses, and the reference image as inputs to generate photorealistic talking faces.

First, we design a universal style encoder to model the motion patterns of facial expressions and head poses in arbitrary

reference style videos. The purpose of the style encoder is to extract the latent style codes (i.e., expression style code or head pose style code) from the sequential 3DMM expression or head pose of the reference style videos. To achieve this, the style encoder utilizes a transformer encoder to study the spatio-temporal co-activation patterns of the input sequential parameters. It then employs a self-attention pooling layer [12] to embed these patterns into the style codes. We also introduce a triplet constraint on the style code space, which allows the universal style encoder to be applied to unseen style clips. Additionally, we observe that the learned style codes lie in a semantically meaningful space.

Afterwards, the unified framework introduces a style-aware decoder that synthesizes stylized animation parameters from audio, based on the style codes. To better incorporate the style codes into the generated animation parameters, the style-aware decoder employs the transformer decoder as the backbone and uses the style code as the query. Leveraging cross attention, the style code can guide the model to focus on closely associating the audio representations with a specific style, thereby enhancing the synthesis of stylized animations. Facial expressions and head movements show distinct motion characteristics, prompting us to develop different decoding strategies. For stylized facial expressions, we propose adaptively generating kernel weights of the feed-forward layers in the transformer decoder conditioned on the style code. This improves lip-sync in various styles and yields more convincing facial expressions. In the stylized head pose decoder, we introduce recurrence into the transformer and predict head movements step-by-step. This allows for the creation of a natural head motion sequence that matches the audio rhythm in different styles.

In summary, our proposed method, StyleTalk++, presents an innovative approach to creating stylized talking videos. Our unified, style-controllable framework enables the extraction and embedding of speaking styles from style reference videos, resulting in the production of natural and photorealistic talking faces in various speaking styles. Extensive experiments demonstrate that our method can generate photorealistic talking faces with diverse speaking styles while satisfying accurate lip synchronization, convincing facial expressions, and natural head movement. We believe that our approach provides a significant contribution to the fields of expressive talking face generation and stylized animation generation.

II. RELATED WORK

A. Audio-Driven Talking Head Generation

With the increasing demand for virtual human creation, driving talking heads with audio [13], [14] has attracted considerable attention. Audio-driven methods can be classified into two categories: person-specific and person-agnostic methods.

1) *Person-Specific Methods*: Person-specific methods [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25] are only applicable to speakers seen during training. [17] produces a high-quality talking video of a target person by using a latent 3D face model. [18] propose a novel text-based talking-head video generation framework that synthesizes high-fidelity facial

expressions and head motions in accordance with contextual sentiments, speech rhythm, and pauses. [22] proposes to convert arbitrary talking head video footage into a normalized space that decouples 3D pose, geometry, texture, and lighting, thereby enabling data-efficient learning and versatile high-quality lip-sync synthesis for video and 3D applications. [23] proposes FACIAL-GAN to jointly learn explicit (facial expression) and implicit (head poses, eye blinks) attributes from audio features. Recently, [24] and [25] introduced neural radiance fields for high-fidelity talking head generation.

2) *Person-Agnostic Methods*: Person-agnostic methods aim to generate talking head videos in a one-shot setting. Early methods [26], [27], [27], [28], [29], [30], [31], [32] focus only on creating accurate mouth movements that are synchronized with the speech content. [31] learns a joint audio-visual representation through audio-visual speech discrimination by associating several supervisions. [32] proposes a novel cascade network structure to reduce the effects of the sound-irrelevant visual dynamics in the image space and explicitly constructs high-level representation from the audio signal and guides video generation using the inferred representation.

With the development of deep learning, a number of methods [1], [2], [4], [5], [6], [33], [34], [35] have been developed to produce more natural talking faces by taking facial expressions and head poses into consideration. [34] proposes a 3D-aware generative network to explicitly model head motion and facial expressions. [6] first simultaneously produces movements of the mouth, eyebrows, and head pose, and then transforms the animation into videos using a flow-guided video generator. [5] implicitly learns pose information directly from reference videos without using intermediate representations. [4] only generates the lip-synced mouth region while using the head pose directly from the original videos. [1] exploits a keypoint-based dense motion field representation to produce natural head motions while keeping non-face regions stable. [2] extracts consistent audio-visual correlations from a specific speaker and achieves satisfactory visual quality and accurate lip-sync. However, although the aforementioned methods can generate videos for arbitrary speakers, none of these methods can create expressive talking head videos.

B. Expressive Talking Head Generation

Although expressive facial expressions are crucial in vivid talking head generation, only a few methods [8], [9], [10], [21], [36], [37], take it into consideration. [8] build emotional talking head dataset MEAD and propose an emotional talking head generation baseline. [21] extract disentangled content and emotional information from audio, and then produce videos guided by the predicted landmarks. However, determining emotions only from audio may lead to ambiguities [10], limiting the applicability of an emotional talking face model. [8], [9] and [38] create emotion-controllable talking faces by employing explicit emotion labels as input, which drop the formulation of personalized differences in speaking styles. [10] and [11] generate expressive talking heads by transferring the expressions in an additional emotional source video to the target speaker frame-by-frame.

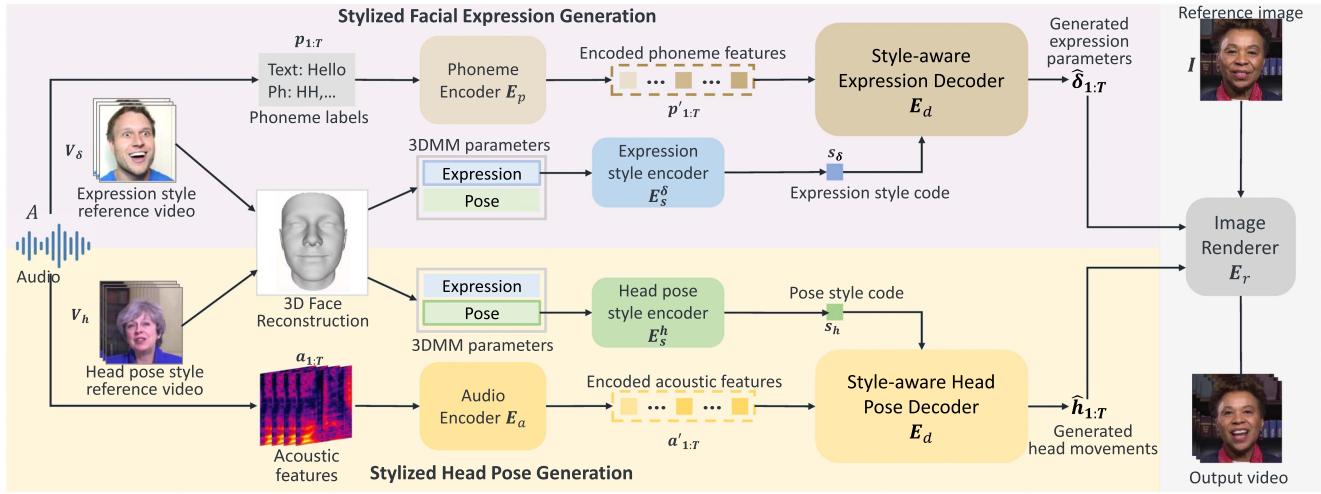


Fig. 1. Illustration of StyleTalk++. Our method can control both facial expression and head pose styles in the generated talking faces using a unified style-controllable framework. These styles can be reflected in two additional style reference videos, including the expression style and head pose style videos, which can be **the same**. The unified style-controllable framework is extended into two branches: (1) The stylized expression generation branch first extracts sequential 3DMM expression parameters from the expression style reference video V_δ using the 3D face reconstruction module, and then feeds them into the expression style encoder E_s^δ to obtain the expression style code s_δ . A phoneme encoder E_p encodes phoneme labels into phoneme features $p'_{1:T}$. Then, the style-aware expression decoder E_d generates the stylized expression parameters $\hat{\delta}_{1:T}$ with s_δ and $p'_{1:T}$. (2) Similarly, the stylized head pose generation branch first extracts sequential head poses from the head pose style reference video and obtains the head pose style code s_h . An acoustic encoder E_a encodes acoustic features into latent features $a'_{1:T}$. Then, we use a style-aware head pose decoder E_d to generate the stylized head movements $\hat{h}_{1:T}$ from s_h and $a'_{1:T}$. Finally, the image renderer E_r takes the assembled $\hat{\delta}_{1:T}$ and $\hat{h}_{1:T}$, and the identity reference image I^r as input, and generates the output video.

To sum up, none of the previous works captures the spatial and temporal co-activations of facial expressions.

C. Audio-Driven Head Movement Generation

Talking head videos with natural head movements appear more realistic. Numerous prior works have explored the generation of audio-driven head movements. One category of methods, such as those presented in [19], [23], [39], [40], only model the audio-to-head-movement mapping reflected in a target speaker video. However, these methods require re-training or finetuning when applied to unseen speakers. Another category of methods, presented in [1], [6], [34], [35], learn the audio-to-head-movement mapping from videos of various speakers and can be applied to unseen speakers without re-training. However, it should be noted that [35] can only produce head movements that slightly swing around the initial pose in the reference image. While [6] and [1] can generate natural head movements from audio, they all have limitations in producing head movements with diverse styles, ignoring the diversity of head pose styles.

D. Extension to Our Prior Work

This paper builds upon our prior research, which includes Audio2Head [1], AVCT [2], and StyleTalk [3]. Our current research extends the idea of controlling StyleTalk in a unified framework that can control both expression and head pose styles. Note that our previous works Audio2Head and AVCT can generate natural head poses and accurate lip-sync, and StyleTalk focuses on generating diverse facial expressions. In this work, we extend this research to explore how to produce accurate lip-sync and natural head poses across diverse styles.

Additionally, we adopt the batched sequential training paradigm proposed in AVCT to achieve realistic talking-head generation. Furthermore, we conduct more comprehensive experiments to validate the effectiveness of our improvements.

III. METHODOLOGY

In this paper, we propose StyleTalk++ for generating the style-controllable talking faces with four inputs: (1) the reference image I^r of the target speaker; (2) the audio clip A of length T providing the speech content; (3) the expression style reference talking video $V_\delta = I_{1:N}^\delta$ of length N , referred to as the expression style clip; (4) the head pose style reference talking video $V_h = I_{1:M}^h$ of length M , referred to as the pose style clip. The expression style clip and pose style clip may be the same. Our method can create photo-realistic talking videos $Y = \hat{I}_{1:T}$ in which the target speaker speaks the speech content with the facial expression style reflected in the expression style clip and head pose style reflected in the head pose style clip.

To generate style-controllable talking faces, we begin by generating the stylized coefficients of a 3DMM, which are then rendered into videos of a specific speaker. We propose a unified style-controllable framework for generating stylized facial expressions and head poses. The framework extracts the speaking style from style reference videos and embeds it into the audio-driven generated coefficients. As shown in Fig. 1, this framework has been extended into two branches for stylized facial expression generation and stylized head pose generation, respectively. Each branch comprises a 3D face reconstruction module, a style encoder, an audio encoder, and a style-controllable decoder. Note that the two branches share the same 3D face reconstruction

module, and their style encoder and audio encoder adopt similar network architectures. Therefore, we first introduce the 3D face reconstruction module in Section III-A and the universal style encoder in Section III-B. We then describe the process of generating stylized head poses and facial expressions in Section III-C and III-D, respectively. Finally, an image renderer is used to convert the generated 3DMM coefficients and reference image into a video. We describe this renderer in Section III-E.

A. 3D Face Reconstruction

With a 3DMM [41], the face shape \mathbf{S} can be represented by an affine model:

$$\mathbf{S} = \mathbf{S}(\delta, \phi) = \bar{\mathbf{S}} + \mathbf{B}_{exp}\delta + \mathbf{B}_{id}\phi, \quad (1)$$

where $\bar{\mathbf{S}}$ is the average face shape, \mathbf{B}_{id} and \mathbf{B}_{exp} are the PCA bases of identity and expression respectively; $\delta \in \mathbb{R}^{64}$ and $\phi \in \mathbb{R}^{80}$ are the corresponding coefficient vectors for a specific 3D face. We adopt the popular 2009 Basel Face Model [42] for $\bar{\mathbf{S}}$ and \mathbf{B}_{id} , and use the expression bases \mathbf{B}_{exp} of [43]. In addition, the head rotation and translation are expressed as $\mathcal{R} \in \mathbb{R}^3$ and $\mathcal{T} \in \mathbb{R}^3$.

An off-the-shelf 3D face reconstruction model [41] is employed for extracting the 3DMM coefficients from portrait images. We employ a subset of 3DMM expression parameters $\delta_{1:N}$ as the facial representation. Given the style clips \mathbf{V}_δ and \mathbf{V}_h , the 3D face reconstruction module extracts the sequential facial expression parameters $\delta_{1:N}$ and head poses $\mathbf{h}_{1:M}$, where $\mathbf{h}_i = \{\mathcal{R}_i, \mathcal{T}_i\}$.

B. Universal Style Encoder

Previous methods for synthesizing stylized facial animations and head pose only transfer the static motions of the static images [10], [11]. Unlike these methods, our approach aims to model the dynamic motion patterns that can guide synthesis. We develop a universal style encoder \mathbf{E}_s to extract the spatio-temporal speaking style reflected in the style clip. The speaking styles in the corresponding sequential facial expression parameters $\delta_{1:N}$ and head poses $\mathbf{h}_{1:M}$ are represented as expression style code s_δ and head pose style code s_h . We use a transformer encoder that takes the sequential $\delta_{1:N}$ or $\mathbf{h}_{1:M}$ as input tokens. The encoder models the temporal correlation between tokens and outputs the style vectors of each token, $s'_{1:N}$. Since the speaking style in a video clip can be identified by a few typical frames, we employ a self-attention pooling layer [12] to aggregate the style information over the style vectors. Specifically, this layer uses an additive attention-based mechanism that computes the token-level attention weights using a feed-forward network. The token-level attention weights represent the frame-level impact on the video-level style code. By summing all the style vectors multiplied by their attention weights, we obtain the final style code $s \in \mathbb{R}^{d_s}$:

$$s = \text{softmax}(W_s H)H^T, \quad (2)$$

where $W_s \in \mathbb{R}^{1 \times d_s}$ is a trainable parameter, $H = [s_1, \dots, s_N] \in \mathbb{R}^{d_s \times N}$ is the sequence of encoded features, d_s is the dimension

of each style vector. Using the same approach, we get the expression style code s_δ and head poses style code s_h .

Our intuition is that the style codes corresponding to similar speaking styles should cluster in the style space. To achieve this, we utilize a triplet constraint on the style codes generated by the style encoder. To apply this constraint, we begin by randomly sampling two additional style clips, \mathbf{V}_c^p and \mathbf{V}_c^n , which reflect similar and dissimilar speaking styles, respectively, to a given style clip \mathbf{V}_c . Corresponding style codes s_c , s_c^p , and s_c^n are then extracted from the triplet-paired videos $\{\mathbf{V}_c, \mathbf{V}_c^p, \mathbf{V}_c^n\}$. Finally, we enforce the constraint on their distances in the style space using the triplet loss [44]:

$$\mathcal{L}_{trip} = \max\{\|s_c - s_c^p\|_2 - \|s_c - s_c^n\|_2 + \gamma, 0\}, \quad (3)$$

where γ is the margin parameter and is set to 5. Note that s_c may represent either head pose or facial expression style code.

C. Stylized Head Pose Generation

1) *Acoustic Encoder*: In the stylized head pose generation branch, the audio is processed into acoustic features, which provide essential rhythm and intonation information related to head movement. As preprocessing, the input raw audio \mathbf{A} is first converted to an acoustic feature sequence $\mathbf{a}_{1:T}$. Each \mathbf{a}_i refers to an acoustic feature frame. To match the video frequency, which is sampled at 25 frames per second, each $\mathbf{a}_i \in \mathbb{R}^{164}$ is composed of acoustic features extracted from four successive sliding windows. A total of 41 acoustic features are extracted from each sliding window, including 13 Mel Frequency Cepstrum Coefficients (MFCCs), 26 Mel-filterbank energy features (FBANK), pitch, and voicelessness. The sliding window has a window size of 25 ms and a step size of 10 ms. We utilize another transformer encoder as our acoustic encoder, denoted as \mathbf{E}_a , to extract acoustic embeddings $\mathbf{a}'_i \in \mathbb{R}^{256}$ from $\mathbf{a}_{1:T}$.

2) *Style-Aware Head Pose Decoder*: The style-aware head pose decoder \mathbf{E}_d^h generates stylized head pose movements by taking sequential acoustic embeddings $\mathbf{a}'_{1:T}$ as input and conditioning on the head pose style code s_h in a sequence-to-sequence manner. However, real-life head movements are non-deterministic and rely not only on the long-term audio rhythm but also on the immediate head pose state. To fit this, we develop our \mathbf{E}_d^h based on Transformer-XL [45], which introduces recurrence into the Transformer by employing hidden states from previous segments as memory for the current segment. By using Transformer-XL, we establish temporal correlations between head movements and audio features, enabling us to produce a head motion sequence that matches the audio rhythm naturally.

We recurrently predict the head movements step by step using \mathbf{E}_d^h , as shown in Fig. 2. At each time step i , we begin by appending the head pose style code s_h with an absolute position embedding [46]. Next, we concatenate the style code with the audio feature \mathbf{a}'_i and feed them into the Transformer-XL. The Transformer-XL combines the input features with memory along the length dimension to calculate the multi-head self-attention. It then outputs the spatial embedding e_i , which encodes the current head pose's spatial state, and updates the memory simultaneously.

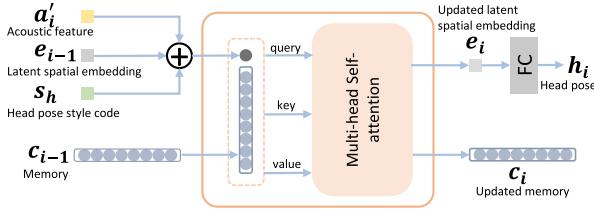


Fig. 2. Illustration of the i -th step of the style-aware head pose decoder. The latent spatial embedding e_i and memory c_i are the intermediate features in this decoder and will be updated at each step.

To account for the local motion state more accurately, we introduce a spatial embedding transition by attaching the previous output spatial embedding e_{i-1} to the integrated input features. This technique enables the decoder to produce more stable head movements and better synchronization with audio. This procedure is formulated as:

$$(c_i, e_i) = \text{TransXL}(c_{i-1}, a'_i \oplus e_{i-1} \oplus s_h), \quad (4)$$

where c_i is the memory of step i in Transformer-XL, \oplus means concatenation, **TransXL** means Transformer-XL.

Finally, we use a fully-connected (FC) layer to decode e_i to head pose $h_i \in \mathbb{R}^6$, where 3 dimensions are for rotation and 3 for translation. Our head motion predictor can handle an arbitrary length of audio input. To ensure better alignment of the generated poses with the camera space of the reference speaker image, the decoder takes an extra initial pose h_r as input. This initial pose can be extracted from the reference speaker image using the 3D face reconstruction model, or it can be a specified head pose. To map h_r to the initial spatial embedding e_0 , we utilize a four-layer feed-forward network. By conditioning on different head pose style codes, our method can create diverse natural-looking head movements while matching the audio rhythm. The stylized head pose generation process can be formulated as :

$$\hat{h}_{1:T} = \mathbf{E}_d^h(a'_{1:T}, s_h, h_r) \quad (5)$$

where $\hat{h}_{1:T}$ means the generated head pose sequence.

3) *Head Pose Objective Function Design*: This section outlines our training approach for the style-aware head pose generation module. Specifically, we jointly train the acoustic encoder \mathbf{E}_a , the head pose style encoder \mathbf{E}_s^h , and the style-aware head pose decoder \mathbf{E}_d^h using the following loss function design:

Head Pose Reconstruction Constraint: As the mapping from audio to head motion is one-to-many, the commonly used L1 and L2 losses are unsuitable for supervising the reconstruction of the head pose. Instead, we use SSIM loss as the reconstruction loss \mathcal{L}_{rec}^h . SSIM (Structural Similarity Index Measure) [47] is a well-known image quality metric that evaluates similarity between two images based on their luminance, contrast, and structure. To be specific, we consider the head motion sequence $h_{1:T} \in \mathbb{R}^{6 \times T}$ as an image of size $6 \times T$, and use SSIM loss to impose structural constraints on it, thereby preserving the velocity, frequency, and amplitude of the head motion. SSIM loss is formulated as:

$$\mathcal{L}_{SSIM} = 1 - \frac{(2\mu\hat{\mu} + C_1)(2\text{cov} + C_2)}{(\mu^2 + \hat{\mu}^2 + C_1)(\sigma^2 + \hat{\sigma}^2 + C_2)}. \quad (6)$$

where $\hat{\mu}$ and $\hat{\sigma}$ are the mean and standard deviation of the generated head pose sequence $\hat{h}_{1:T}$, and μ and σ are that of the ground truth head pose sequence. cov is the covariance. C_1 and C_2 are two small constants. To train our model to reconstruct the head pose, we utilize SSIM loss as the reconstruction loss \mathcal{L}_{rec}^h .

Head Pose Temporal Discriminator: In order to improve the smoothness of the generated head pose sequence $\hat{h}_{1:T}$, we use a head pose temporal discriminator \mathbf{D}_{tem}^h , which learns to differentiate between real and fake input head pose sequence. Specifically, we modify the 2D PatchGAN discriminator [48], [49], [50], [51], [52], [53], which is designed to process patches of the input image rather than the entire image, into a 1D window discriminator that focuses on the temporal window of the input sequence. Following the same network structure as the vanilla PatchGAN discriminator, our \mathbf{D}_{tem}^h performs 1D convolution instead of 2D convolution on the input head pose sequence along the temporal axis. This helps to classify whether 70×70 overlapping head pose windows are real or fake and in turn, improves the smoothness of the generated head pose sequence. Additionally, we employ LSGAN [54] to calculate the adversarial loss:

$$\mathcal{L}_{tem}^h = \|\mathbf{D}_{tem}^h(\hat{h}_{1:T}) - 1\|_2. \quad (7)$$

Head Pose Style Discriminator: To ensure consistency in the style between the generated head movements and the specified head pose style, we propose a head pose style discriminator \mathbf{D}_{style}^h , which shares a similar network architecture to \mathbf{D}_{tem}^h . The primary objective of \mathbf{D}_{style}^h is to distinguish whether the input head poses belong to the specified head pose style. \mathbf{D}_{style}^h takes the integrated head pose sequence and style code as input. Specifically, the style code is first repeated T times and appended with position embeddings, and then concatenated with the head pose sequence along the time dimension. During training, \mathbf{D}_{style}^h learns to minimize the following objective:

$$\begin{aligned} \mathcal{L}_{style}^{h,D} = & \|\mathbf{D}_{style}^h(h_{1:T}, s_h) - 1\|_2 \\ & + \|\mathbf{D}_{style}^h(h_{1:T}, s_h^n) - 0\|_2, \end{aligned} \quad (8)$$

where s_h denote the head pose style reflected in the ground truth head pose $h_{1:T}$, s_h^n denotes another head pose style that is not similar to s_h . When training the stylized head pose generation module, the same speech input is used along with different style codes. To ensure that the style of the generated head movements aligns with the specified head pose style, we utilize a style adversarial loss. The style adversarial loss is defined as:

$$\begin{aligned} \mathcal{L}_{style}^h = & \|\mathbf{D}_{style}^h(\mathbf{E}_d^h(a'_{1:T}, s_h, h_r), s_h) - 1\|_2 \\ & + \|\mathbf{D}_{style}^h(\mathbf{E}_d^h(a'_{1:T}, s_h^n, h_r), s_h^n) - 0\|_2. \end{aligned} \quad (9)$$

Furthermore, because all style reference lengths are randomly sampled during the training process, these lengths may differ from that of the generated head pose, which is determined by the length of the input audio. This indicates that the generated head pose cannot be a mere replication of the reference when style adversarial loss ensures that the generated head movements conform to the specified head pose style.

Full Objective: Our full objective for the stylized head pose generation module is:

$$\mathcal{L}_{total}^h = \lambda_{rec}^h \mathcal{L}_{rec}^h + \lambda_{trip}^h \mathcal{L}_{trip}^h + \lambda_{tem}^h \mathcal{L}_{tem}^h + \lambda_{style}^h \mathcal{L}_{style}^h, \quad (10)$$

where \mathcal{L}_{trip}^h is the triplet loss for the head pose style code, as introduced in Section III-B, and the hyper-parameters λ_{rec}^h , λ_{trip}^h , λ_{tem}^h and λ_{style}^h are set to 100, 1, 10, 10 respectively. Note that the length of the input and output sequences is set to 256 during training, but can be of any length during inference.

D. Stylized Facial Expression Generation

1) *Phoneme Encoder:* In the stylized facial expression generation module, we intend to merely extract articulation-related information from the audio. This will eliminate any interference that may affect the speaking style of the generated facial expressions, such as emotion and intensity. To achieve this, we utilize phoneme labels instead of acoustic features to represent the audio signals. The phoneme labels $p_{1:T}$ are subsequently transformed into phoneme embeddings and fed into the phoneme encoder E_p , which produces sequential articulation representations $p'_{1:T}, p'_t \in \mathbb{R}^{256}$. The phoneme encoder comprises a vanilla transformer encoder, and the phoneme labels are extracted using a speech recognition tool.

2) *Style-Aware Facial Expression Decoder:* At the early stage, we employ the vanilla transformer decoder as the facial expression decoder E_d^δ , which takes the articulation representations $p'_{t-w:t+w}$ and the facial expression style code s_δ as input. Specifically, we repeat the expression style code $2w+1$ times and then add them with positional encodings to obtain the style tokens. The style tokens serve as the query of the transformer decoder, and the latent articulation representations serve as the key and value. The middle output token is fed into a feed-forward network to generate the output expression parameter $\hat{\delta}_t$. The facial expression generation process can be formulated as:

$$\hat{\delta}_{1:T} = E_d^\delta(p'_{1:T}, s_\delta) \quad (11)$$

When utilizing the aforementioned decoder, we observe defective lip movements and facial expressions when generating stylized talking faces with large facial movements. Inspired by [55] and [56], we assume that the static kernel weights cannot model the diverse speaking styles. With this assumption, we design a style-aware adaptive transformer, which dynamically adjusts the network weights according to the style code, as shown in Fig. 3. Specifically, since [57] reveals that the feed-forward layers play the most important role in transformer decoder, we replace the feed-forward layers with novel style-aware adaptive feed-forward layers. The style-aware adaptive layer utilizes $K = 8$ parallel sets of weights \tilde{W}_k, \tilde{b}_k . Such parallel weights are expected to be the experts for modeling the distinct facial motion patterns of the different speaking styles. Then we introduce the additional layers followed by Softmax to adaptively compute the attention weights over each set of weights depending on the style code. Then the feed-forward layer weights are aggregated

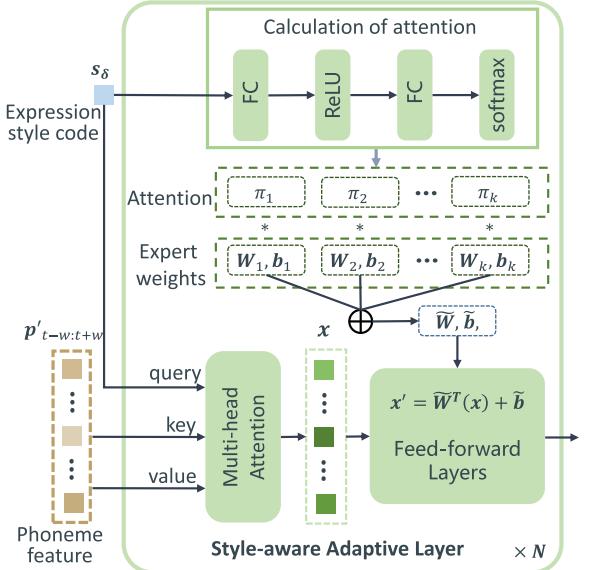


Fig. 3. Illustration of the style-aware adaptive transformer decoder layer.

dynamically via the attention weights:

$$\tilde{W}(s_\delta) = \sum_{k=1}^K \pi_k(s_\delta) \tilde{W}_k, \quad \tilde{b}(s_\delta) = \sum_{k=1}^K \pi_k(s_\delta) \tilde{b}_k, \\ \text{s.t. } 0 \leq \pi_k(s_\delta) \leq 1, \quad \sum_{k=1}^K \pi_k(s_\delta) = 1, \quad (12)$$

where π_k is the attention weight for k^{th} feed-forward layer weights \tilde{W}_k, \tilde{b}_k . The output of style-controllable dynamic feed-forward layers is then obtained by:

$$y = g\left(\tilde{W}^T(s_\delta)x + \tilde{b}(s_\delta)\right), \quad (13)$$

where g is an activation function. Our experiments show that the style-controllable dynamic decoder helps to create accurate stylized lip movements and natural stylized facial expressions in diverse speaking styles.

3) *Disentanglement of Upper and Lower Faces:* In our experiments, we observed that the upper face and the lower face exhibit different motion patterns and have distinctive correlations with the audio input. Specifically, the upper face (eye, eyebrow) moves at a low frequency, while the lower face (mouth) moves at a high frequency. Thus, it is reasonable to model the motion patterns of the two parts with separate networks.

To begin, we divided the expression parameters into two groups: The lower face group and the upper face group. We then utilized two parallel style-controllable dynamic decoders, namely the upper face decoder and the lower face decoder, to generate the corresponding group of express parameters. For the lower face group, we selected 13 out of the 64 expression parameters that are highly related to mouth movements. For the upper face group, we used the remaining parameters. Finally, we concatenated the two groups of generated expression parameters to obtain the final generated expression parameters.

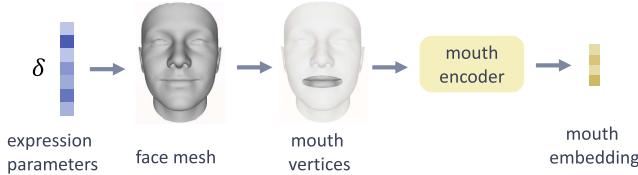


Fig. 4. Mouth embedding extraction in lip-sync discriminator.

4) *Facial Expression Objective Function Design*: Because the stylized facial expression generation module generates each frame individually, we adopt a batched sequential training strategy [2] to improve the temporal consistency. Specifically, we generate successive $L = 64$ frames $\delta_{1:L}$ at one time as a clip. Specifically, we jointly train the phoneme encoder E_p , the facial expression style encoder E_s^δ , and the style-aware facial expression decoder E_d^δ using the following loss function design:

Facial Expression Reconstruction Constraint: During training, we reconstruct the facial expressions of each clip in the self-driven setting. We adopt a combination of the L1 loss and SSIM loss:

$$\mathcal{L}_{rec}^\delta = \mu \mathcal{L}_{L1}(\delta_{1:T}, \hat{\delta}_{1:T}) + (1 - \mu) \mathcal{L}_{ssim}(\delta_{1:T}, \hat{\delta}_{1:T}), \quad (14)$$

where $\delta_{1:T}$ and $\hat{\delta}_{1:T}$ are the ground truth and reconstructed facial expressions respectively. μ is a ratio coefficient and is set to 0.1.

Lip-sync Discriminator: Due to the variability in mouth shape across different speaking styles, achieving accurate lip synchronization is an extremely challenging task. Inspired by SyncNet [4], we design a lip-sync discriminator D_{sync} , which is trained to discriminate the synchronization between audio and mouth by randomly sampling an audio window that is either synchronous or asynchronous with a video window.

We have made modifications to the original SyncNet to enhance the lip sync of synthetic facial expressions. Since the 3DMM expression PCA bases controlling mouth movements also affect other facial movements, we first convert expression parameters into a face mesh using the PCA expression bases, and then extract the mouth vertices as a pure mouth shape representation, as illustrated in Fig. 4. We specifically select 404 vertices located in the mouth area of face meshes in the 3D morphable model. We input the mesh vertex coordinates into the mouth encoder and phonemes into the audio encoder, instead of feeding images and acoustic features into the original SyncNet.

Specifically, we use PointNet [58] as the mouth encoder to extract the mouth embedding e_m , and another phoneme encoder to compute the audio embedding e_a from the phoneme window. We adopt cosine similarity to indicate the probability that e_m and e_a are synchronous [4]:

$$P_{sync} = \frac{e_m \cdot e_a}{\max(\|e_m\|_2 \cdot \|e_a\|_2, \epsilon)}, \quad (15)$$

where ϵ is a small constant. The expression generation module maximizes the synchronous probability via a sync loss \mathcal{L}_{sync} on each frame of the generated clip:

$$\mathcal{L}_{sync} = \frac{1}{L} \sum_{i=1}^L -\log(P_{sync}^i). \quad (16)$$

Facial Expression Style Discriminator: The facial expression style discriminator D_{style}^δ is designed to classify the speaking style of the input sequential 3DMM expression parameters $\delta_{1:L}$. Specifically, the style discriminator generates a probability distribution $P^s \in \mathbb{R}^C$, indicating the likelihood that the sequence of parameters belongs to each speaking style. Here, C is the number of speaking styles. The style discriminator follows the PatchGAN structure, and is initially pre-trained on a dataset containing C speaking styles using cross-entropy loss. Once pre-trained, the style discriminator is frozen and used to guide the generative modules towards producing vivid speaking styles via a style loss function $\mathcal{L}_{style}^\delta$:

$$\mathcal{L}_{style}^h = -\log(P_i^s), \quad (17)$$

where i is the category of the ground-truth speaking style of the facial expressions.

Facial Expression Temporal Discriminator: To improve the temporal stability of the generated facial expressions, we utilize a facial expression temporal discriminator D_{tem}^δ , similar to the head pose temporal discriminator D_{tem}^h , to calculate the adversarial loss \mathcal{L}_{tem}^δ .

Full Objective: Our full objective for training the stylized facial expression generation module is given by a combination of the aforementioned loss terms:

$$\begin{aligned} \mathcal{L}_{total}^\delta = & \lambda_{rec}^\delta \mathcal{L}_{rec}^\delta + \lambda_{trip}^\delta \mathcal{L}_{trip}^\delta + \lambda_{sync} \mathcal{L}_{sync} \\ & + \lambda_{tem}^\delta \mathcal{L}_{tem}^\delta + \lambda_{style}^\delta \mathcal{L}_{style}^\delta, \end{aligned} \quad (18)$$

where $\mathcal{L}_{trip}^\delta$ is the triplet loss for the facial expression style code, as introduced in Section III-B, and we use $\lambda_{rec}^\delta = 88$, $\lambda_{trip}^\delta = 1$, $\lambda_{sync} = 1$, $\lambda_{tem}^\delta = 1$ and $\lambda_{style}^\delta = 1$.

E. Image Render

After obtaining the generated stylized head movements $\hat{h}_{1:T}$ and stylized facial expressions $\hat{\delta}_{1:T}$, we integrate them and feed them into an image renderer E_r along with the reference one-shot image to produce the final output videos. Our image renderer follows the network architecture of PIRenderer [59], which is capable of generating photo-realistic results with accurate motions by utilizing a source portrait image and target 3DMM parameters. PIRenderer comprises a mapping network, a warping network, and an editing network. The mapping network produces latent vectors from the 3DMM parameters. Instructed by the latent vectors, the warping network estimates the dense motion field between the source and desired images, producing a coarse image with the estimated deformations. Finally, the editing network generates the final images from the coarse images.

IV. IMPLEMENTATIONS

A. Datasets

In this paper, we use four widely-used talking face datasets: VoxCeleb [60], HDTF [6], MEAD [8], and HeadMotion [61]. All videos are aligned by cropping and resizing to 256×256 , as

done in [62]. The videos are sampled at 25 FPS, and the audio is pre-processed to 16 KHZ.

VoxCeleb Dataset: VoxCeleb is an audio-visual dataset that consists of short clips of human speech recorded in the wild. It comprises utterances from more than 1,000 speakers of different ethnicities, accents, professions, and ages.

HDTF Dataset: HDTF consists of 362 high-quality videos of over 300 subjects. The resolution of original videos is 720P or 1080P. The test set comprises 20 videos, totaling around 10K frames.

MEAD Dataset: Mead is a high-quality emotional talking-face dataset recorded in the lab. It includes videos in which different speakers speak with eight different emotions at three different intensity levels. Here, we have selected 42 actors for training and 6 actors for testing.

HeadMotion Dataset: HeadMotion is a recently built head motion dataset collected from the internet. This dataset consists of 751 single-person talking videos, each recorded without camera movements. Various types of head movements are contained within this dataset.

B. Implementation Details

Style encoder: The universal style encoder \mathbf{E}_s takes as input the sequential expression parameters $\delta_{1:N}$ or head poses $\mathbf{h}_{1:M}$. The length N of the expression style reference video is from 64 to 256 (about 2 ~ 10 seconds) and the length M of the head pose style reference video is from 128 to 512 (about 5 ~ 20 seconds). We increase their dimension to 256 by feeding them into a linear layer. Next, we feed the sequence features into a transformer encoder, which contains 3 8-head transformer encoder layers with a hidden size of 256. The output tokens are then aggregated by a self-attention pooling layer, as introduced in Section III-B, to obtain the final style code s .

Acoustic encoder and phoneme encoder: Both \mathbf{E}_a and \mathbf{E}_p employ the same transformer encoder architecture as that used in the universal style encoder \mathbf{E}_s . In \mathbf{E}_a , the dimension of each frame's acoustic feature is first increased to 256 by a feed-forward layer. As phonemes are denoted as discrete labels, each phoneme label is mapped into a word embedding of 256 in \mathbf{E}_a .

Head pose decoder: We generate one frame of head pose in each step in the style-aware head pose decoder \mathbb{E}_d^h . At each time step i , the concatenated features of a'_i , e_{i-1} , and s_h are fed to a feed-forward layer to decrease the feature dimension to 256, and they are then appended to the end of the memory tokens. These tokens are fed to a 2-layer transformer, where the head number is 8 and the hidden size is 256. The last output token is used as e_i , which is fed to a fully connected layer to produce the current head pose \mathbf{h}_i . The length of the memory is set to 128, and we truncate the last 128 tokens from the output tokens to form the new memory.

Expression decoder: Our style-aware expression decoder \mathbb{E}_d^δ is implemented based on the transformer decoder, where the style tokens serve as queries and the audio features serve as keys and values. The transformer decoder has three 8-head decoder layers and the hidden dimension is 256. In this decoder,

we replaced the feed-forward layers with style-aware adaptive feed-forward layers. For these layers, we initialize eight sets of weights for a feed-forward layer with a hidden size of 2048. The style code is then fed into a linear attention network to obtain eight attention weights, which are used to aggregate the aforementioned weights. The attention network comprises two fully connected layers and a Softmax layer, with a hidden dimension of 64 for the two fully connected layers. Finally, the eight sets of weights are aggregated using the attention weights to obtain the final weights for the feed-forward layer.

Image renderer: We used the well-known PIRenderer [59] as the network architecture for our image renderer. For detailed information on the architecture, please refer to [59]. In this paper, we trained an off-the-shelf image renderer on three talking face datasets, namely HDTF [6], VoxCeleb [60], and MEAD [8].

Temporal discriminator and style discriminator: Our head pose temporal discriminator \mathbf{D}_{tem}^h , facial expression temporal discriminator \mathbf{D}_{tem}^δ , head pose style discriminator \mathbf{D}_{style}^h , and facial expression style discriminator $\mathbf{D}_{style}^\delta$ have similar network architectures. We replaced the 2D convolutions in the vanilla 70×70 PatchGAN [49] with 1D convolutions and performed convolution operations along the temporal dimension. The discriminator judges whether each 70-length input sequence is real or fake.

C. Training Details

We use PyTorch [63] to implement our method. Our framework is implemented by Pytorch [63]. We employ Adam optimizer [64] for all training.

The branch responsible for generating stylized head poses jointly trains models \mathbf{E}_s^h , \mathbf{E}_a , \mathbf{E}_d^h , and \mathbf{D}_{tem}^h with an initial learning rate of 1e-4, which decays to 2e-6 within 100 epochs. Specifically, during the first 50 epochs, \mathbf{D}_{style}^h is not involved in the training, but is then jointly trained with other modules in the following 50 epochs. These models are trained on a combination of the HDTF and HeadMotion datasets. We observe that individuals maintain a consistent head movement style over time. Therefore, we sample triplet-paired samples according to the following strategy for the triplet constraint (Section. III-B) of the head pose style encoder during training. Specifically, head pose style video clips sampled from segments near the anchor are considered to have the same style as the anchor, while clips sampled from non-adjacent segments have a different style.

To train the modules in the stylized facial expression generation branch, we first construct our dataset based on MEAD and HDTF. For MEAD, we assume that video clips where the speaker expresses the same emotion at the same intensity level share the same expression style. For HDTF, we assume that video clips from one speaker share the same speaking style. We obtain 1,104 speaking styles, and each style corresponds to a set of videos in the training set. For the triplet constraint of the expression style encoder, positive samples are sampled from the same set as the given one, while negative samples are sampled from any other set. \mathbf{D}_{sync} and $\mathbf{D}_{style}^\delta$ are trained on this dataset with a learning rate of 0.0001. Afterward, \mathbf{D}_{sync} and $\mathbf{D}_{style}^\delta$ are frozen, and \mathbf{E}_a ,

\mathbf{E}_s , \mathbf{E}_d , and \mathbf{D}_{tem} are jointly trained for 50 epochs with a learning rate of 0.0001.

D. Metrics

In this paper, we use several widely adopted metrics to evaluate the effectiveness of the proposed methods. For evaluating lip synchronization, we use the confidence score of SyncNet [65] ($\text{Sync}_{\text{conf}}$) and the Landmark Distance around the mouth (**M-LMD**) [32]. To assess the accuracy of generated facial expressions, we use the Landmark Distance on the whole face (**F-LMD**). To evaluate the quality of generated talking head videos, we adopt **SSIM** and the Cumulative Probability of Blur Detection (**CPBD**) [66]. Furthermore, we separately evaluate the quality of the generated head pose using **SSIM** and peak signal-to-noise ratio (**PSNR**).

V. EXPERIMENTS

In this section, we perform extensive experiments to validate our proposed method. We conduct quantitative evaluations with SOTA in Section V-B and qualitative evaluations in Section V-A. In Section V-C, we conduct ablation studies to validate the effectiveness of each component in our stylized head pose generation branch and stylized facial expression generation branch. We also conduct experiments in Section V-D to inspect the learned style code space. Furthermore, in Section V-E, we perform a user study. Finally, in Section V-F, we provide a discussion. To present and compare the results more clearly, most of the experiments in this section are divided into two groups to assess the overall visual effects and the head motion effects separately. Specifically, for evaluating the visual effects, we fix the head motion style, while for evaluating the head motion effects, we fix the facial expression style.

A. Quantitative Evaluation

To comprehensively evaluate our method, we conduct two sets of quantitative experiments. The first set focuses on evaluating the visual quality and facial expressions, including lip sync, of the generated videos. The second set focused on evaluating the quality of the generated head poses.

1) *Evaluation on Visual Effect:* We first conduct quantitative evaluations on the overall quality of the generated talking videos. We compare our method with state-of-the-art talking face methods, including MakeitTalk [35], Wav2Lip [4], PC-AVS [5], AVCT [2], GC-AVT [11], and EAMM [10]. The experiments are performed in the self-driven setting on the test set of MEAD and HDTF, where the speaker and speaking style are not seen during training. We select the first image of each video as the reference image and use the corresponding audio clip as the audio input. For all methods, poses are derived from ground truth videos. However, Wav2Lip can only generate mouth area movements, so the head poses are fixed in its samples. EAMM, GC-AVT and our method require an additional expression reference video as input. For these methods, the ground truth videos are used as the expression reference videos. The compared methods' samples are generated using either their released codes or with the help

of their authors. Table I reports the results of the quantitative evaluation.

Our method outperforms most other methods in terms of various metrics on both MEAD and HDTF datasets. Since Wav2Lip merely generates mouth movements and does not change other parts of the reference images, it obtains the highest CPBD score on MEAD. However, the mouth area generated by Wav2Lip is blurry (See Fig. 5). Since Wav2Lip is trained using SyncNet as a discriminator, it is reasonable for Wav2Lip to obtain the highest confidence score of SyncNet ($\text{Sync}_{\text{conf}}$) on MEAD. The score is even higher than that of the ground truth. Our method achieves the closest $\text{Sync}_{\text{conf}}$ scores to ground truth on MEAD and the highest on HDTF, indicating our method's ability to produce precise lip-sync. In terms of M-LMD metric, our method achieves the best scores, further demonstrating the accuracy of our lip-sync generation. Moreover, our method performs the best under the F-LMD metric, demonstrating our method's ability to generate facial expressions that match the reference speaking style. Therefore, our method outperforms other methods in generating high-quality lip-sync with accurate facial expressions across various metrics.

2) *Evaluation on Head Movements:* We then conduct another quantitative evaluation on the generated head poses by comparing our method with MakeitTalk, Audio2Head [1], and PHM [61]. These methods are capable of generating head movements from audio. The experiments are also performed in the self-driven setting on the test sets of HDTF and HeadMotion. In each case, we selected the first image of each video as the reference image and used the corresponding audio clip as the input. For PHM and our method, we used the ground truth video as the source of the head pose style. For MakeitTalk, Audio2Head, and PHM, we extract the head pose sequence from their generated videos. As can be seen in Table II, our method achieved the best scores on all metrics on HDTF and HeadMotion datasets.

B. Qualitative Evaluation

In addition to the person-agnostic methods, we also conduct qualitative evaluations using person-specific methods to demonstrate the superiority of our proposed method.

1) *Comparison With Person-Agnostic Methods:* We conduct two sets of qualitative evaluations to assess the overall visual effect and generated head poses independently when compared to person-agnostic methods. In the first set, we compare our method with speaker-agnostic (one-shot) methods, including Wav2Lip, PC-AVS, AVCT, EAMM, and GC-AVT, to visualize the video quality, lip-sync, and facial expression. Fig. 5 displays the results of this comparison. Note that for EAMM, GC-AVT, and our method, we use the same video clip as the expression style reference. For PC-AVS, AVCT, EAMM, GC-AVT, and our method, we extract the head poses from the Mouth-GT videos where the audio input comes from. For the second set of qualitative evaluations, we conduct a comparison with Audio2Head and MakeitTalk to visualize the generated head poses. Fig. 6 displays the results of this comparison, where we selected a neutral talking video as the expression style reference for our

TABLE I
QUANTITATIVE RESULTS ON THE VISUAL EFFECTS OF THE GENERATED VIDEOS ON MEAD AND HDTF DATASET

Method	MEAD					HDTF				
	SSIM↑	CPBD↑	F-LMD ↓	M-LMD ↓	Sync _{conf} ↑	SSIM↑	CPBD↑	F-LMD ↓	M-LMD ↓	Sync _{conf} ↑
MakeitTalk	0.725	0.106	3.969	5.324	2.104	0.593	0.248	5.084	4.447	2.563
Wav2Lip	0.795	0.178	2.718	4.052	5.257	0.618	0.299	4.544	3.630	3.072
PC-AVS	0.504	0.071	5.828	4.970	2.183	0.422	0.132	10.506	3.931	2.701
AVCT	0.832	0.139	2.923	5.520	2.525	0.755	0.233	2.733	3.610	3.147
GC-AVT	0.340	0.142	8.039	7.103	2.417	0.337	0.296	10.537	6.206	2.772
EAMM	0.397	0.084	6.698	6.478	1.405	0.387	0.144	7.031	6.857	1.799
Ground Truth	1	0.222	0	0	4.131	1	0.307	0	0	3.961
Ours	0.837	0.164	2.122	3.249	3.474	0.812	0.302	1.941	2.412	3.165

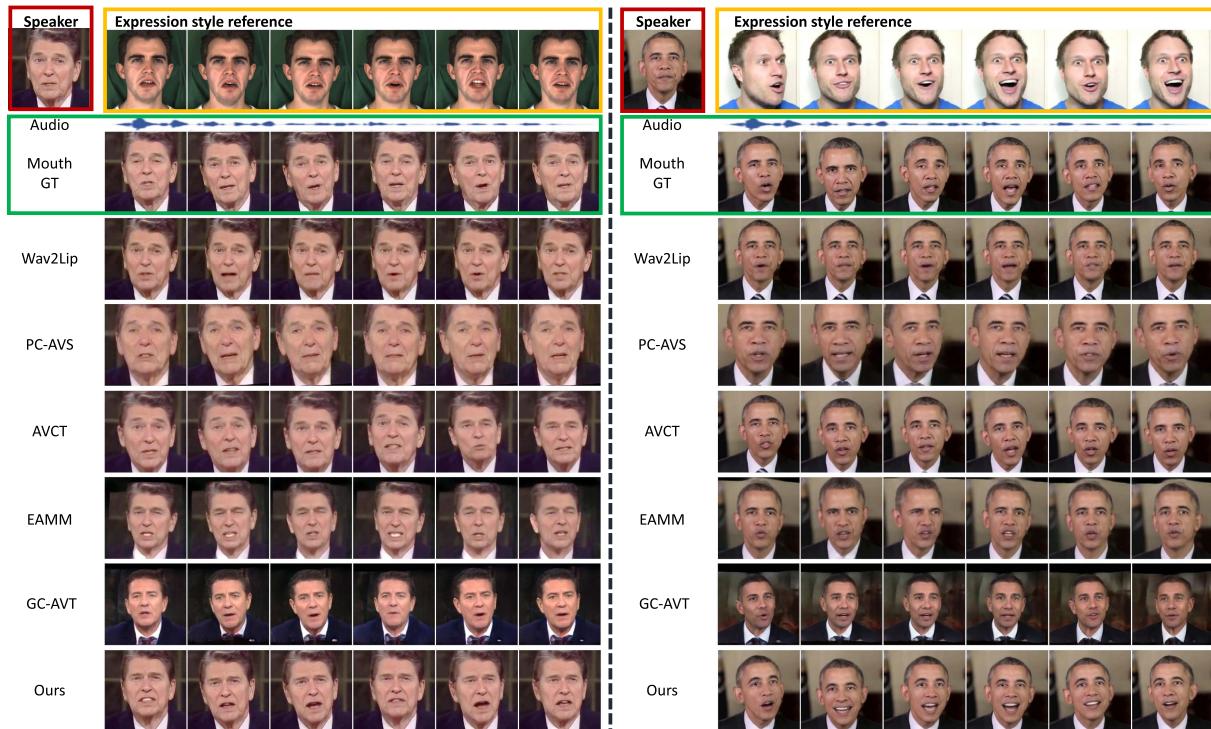


Fig. 5. Qualitative comparisons with the person agnostic methods. The identity reference, expression style reference videos, and audio-synced videos are displayed in the first two rows. This figure mainly showcases comparisons in visual quality, facial expression, and lip-sync accuracy. It is worth noting that for EAMM, GC-AVT, and our method, we use the same video clip as the expression style reference. For PC-AVS, AVCT, EAMM, GC-AVT, and our method, head poses are derived from the Mouth GT video. Please zoom in or see our demo video for more details.

TABLE II
QUANTITATIVE RESULTS OF GENERATED HEAD MOVEMENTS ON HDTF AND HEADMOTION DATASET

Method	HDTF		HeadMotion	
	SSIM↑	PSNR↑	SSIM↑	PSNR↑
MakeitTalk	0.747	25.49	0.646	24.66
Audio2Head	0.761	26.52	0.707	26.11
Ours	0.847	27.58	0.786	26.75

method. It is important to note that in both sets of evaluations, the identity reference, style reference, and audio were all unseen during training.

As shown in Fig. 5, our method can generate talking faces that accurately match a reference expression style, while achieving

precise lip-sync and better preservation of the speaker's identity (please refer to our demo video). Among all the methods, only EAMM, GC-AVT, and our method can perform expression style control. However, EAMM and GC-AVT can only control expression styles in the upper face, such as the eyes and eyebrows, while failing to control the stylized shape of the mouth. Moreover, the expression styles of videos generated by these methods are significantly inconsistent with those of the style reference. In terms of lip-sync, only Wav2Lip, AVCT, PC-AVS, and GC-AVT are competitive with our method. However, they only model one neutral speaking style in the mouth area, making them unable to produce natural lip-sync in various styles. Furthermore, GC-AVT is unable to preserve the speaker's identity well, and both EAMM and GC-AVT are incapable of producing realistic backgrounds. In contrast, our method can imitate speaking styles in the entire face from arbitrary style clips while achieving

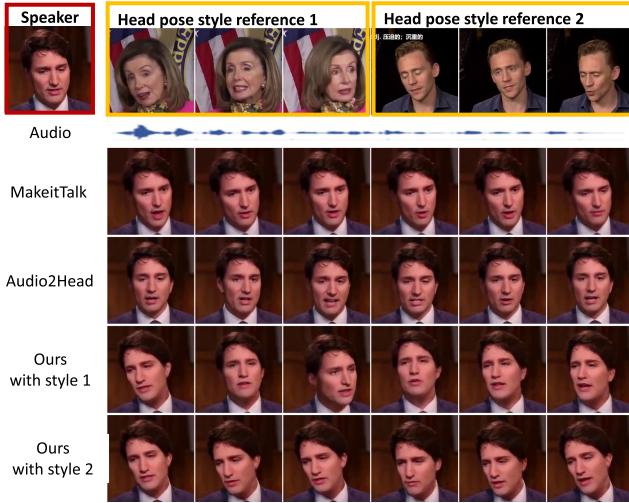


Fig. 6. Qualitative comparisons with the person agnostic methods that are capable of synthesizing head poses from audio. This figure primarily demonstrates comparisons of generated head movements. For our method, we use two video clips as the head pose style references. Style 1 presents a motion pattern of frequent left and right head shaking, while Style 2 mainly looks to the right when speaking.

accurate lip-sync, preserving speaker identity, and generating plausible backgrounds.

Based on Fig. 6, it is evident that our method is capable of extracting distinct head motion patterns from the reference video and generating diverse head motion sequences using the same audio input, under the guidance of the extracted head motion patterns. Furthermore, the different head pose sequences generated are synchronized with the rhythm of the same input audio. However, Audio2Head can only generate natural head movements and cannot control the style of the synthesized head motions. As for MakeItTalk, it can only generate slight movements that swing around the initial head pose in the reference image. Another noteworthy aspect is that we are able to apply head motion styles obtained from different camera planes to the reference image in its own plane of space. For example, in Fig. 6, the speaker in the *head pose reference style 2* is farther away from the camera than the subject in the reference image, and our method predicts the stylized head movements from the perspective of the reference image camera. This also indicates that our method indeed extracts head motion patterns from the reference video rather than simply transferring head movements.

2) *Comparison With Person-Specific Methods:* We further compare our method with person-specific emotional talking face methods, including Write-a-Speaker [18] and EVP [21]. For both methods, we crop video clips from their demo videos. Then we select one neutral image as the reference image and a video in MEAD with the same emotion as the style clip. The qualitative results are shown in Fig. 7. Compared with the other two works, our method also generates vivid emotional facial expressions and achieves comparable lip-sync. Note that our method is based on the one-shot setting, while the other two methods are trained on a long reference video of the target speaker.



Fig. 7. Qualitative comparisons with person-specific methods.

TABLE III
QUANTITATIVE RESULTS OF THE ABLATION STUDY ON THE STYLIZED HEAD POSE GENERATION BRANCH ON THE HEADMOTION DATASET

Method	SSIM↑	PSRN↑
w/ SDT-TF	0.566	23.56
w/ SDT-RR	0.643	24.86
w/o SET	0.629	24.15
w/o D_{style}^h	0.736	25.78
w/o \mathcal{L}_{trip}^h	0.683	25.10
w/o D_{tem}^h	0.710	25.52
w/ Ph	0.593	24.22
Full	0.786	26.75

C. Ablation Study

1) *Ablation Study on Stylized Head Pose Generation:* We first conduct ablation studies on the components in the stylized head pose generation modules on the HeadMotion dataset. Specifically, we analyze the impact of removing individual modules from the overall system. We design eight variants: (1) replace the Transformer-XL with the standard transformer decoder (**w/ SDT-TF**) and train the model with teacher-forcing strategy, (2) replace the Transformer-XL with the standard transformer decoder (**w/ SDT-RR**) and train the model using recurrence strategy, (3) remove the spatial embedding transition by eliminating the input of e_{i-1} at each time step i (**w/o SET**), (4) remove the style discriminator (**w/o D_{style}^h**), (5) remove triplet loss (**w/o \mathcal{L}_{trip}^h** , note that we also remove D_{style}^h in this variant), (6) remove the temporal discriminator (**w/o D_{tem}^h**), (7) use phonemes as input instead of acoustic features (**w/ Ph**), and (8) our full model (**Full**). The results are shown in Table III and Fig. 9. Please see our demo video for more details on the dynamics of the generated head movements.

An interesting observation is that when we replace the Transformer-XL with the standard transformer (**w/ SDT-TF**), the network tends to only produce minimal head movements around the initial position, indicating that the recurrence is essential for enhancing the dynamism of head movement generation. Furthermore, despite both **w/ SDT-TF** and **w/o SET** applying the recurrence strategy, neither explicitly inputs the spatial head pose embedding from the previous moment to the



Fig. 8. Qualitative results of the ablation study on stylized expression generation branch.

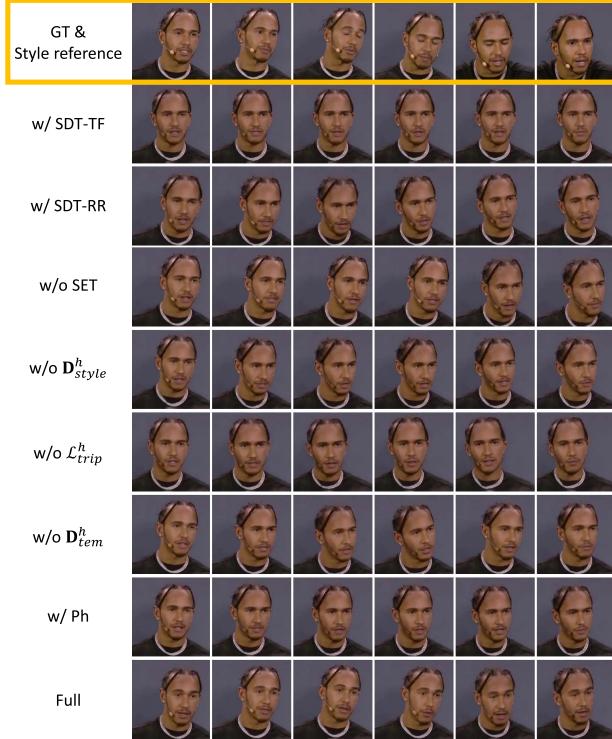


Fig. 9. Qualitative results of the ablation study on stylized head pose generation branch.

current step, resulting in highly unstable head movements. This implies that subsequent movements depend on both the current speech and the current head movements. Even though both **w/o D_{style}^h** and **w/o D_{temp}^h** can produce natural head movements, they achieve lower scores than our full model. This suggests that the triplet constraint and the style discriminator make our model sensitive to head motion patterns. Without using D_{temp}^h , the generated head movements will show jitters. Moreover, when we replace the input acoustic features with phonemes, our model tends to produce head movements with smaller dynamics and loses rhythm corresponding with audio. This implies that incorporating acoustic features enhances the performance of the generated head motion. Therefore, these results indicate that each component in the stylized head pose generation modules contributes significantly to the improvements of the final results.

2) Ablation Study on Stylized Facial Expression Generation Modules: We then conduct ablation studies on the components

TABLE IV
QUANTITATIVE RESULTS OF THE ABLATION STUDY ON THE STYLIZED EXPRESSION GENERATION BRANCH ON THE MEAD DATASET

Method	SSIM↑	CPBD↑	F-LMD ↓	M-LMD ↓	Sync _{conf} ↑
w/o StyQ	0.836	0.161	2.403	3.651	3.455
w/o DyFFN	0.830	0.165	2.414	4.178	3.059
K = 4	0.831	0.163	2.327	3.524	3.331
K = 16	0.835	0.161	2.133	3.396	3.473
w/o D_{style}^{δ}	0.836	0.160	2.483	3.628	3.430
w/o L_{trip}^{δ}	0.837	0.160	2.401	3.771	3.532
w/o D_{sync}	0.834	0.164	2.281	4.351	2.305
Full (K = 8)	0.837	0.164	2.122	3.249	3.474

in the stylized facial expression generation modules on the MEAD dataset. Similarly, we analyze the impact of removing individual modules from the overall system or altering some settings. We design eight variants: (1) use the phoneme representation as the query in the decoder, rather than the style code (**w/o StyQ**), (2) replace the adaptive feedforward layer with the vanilla feedforward layer (**w/o DyFFN**), (3) set $K = 4$ in dynamic feedforward layer ($K = 4$), (4) set $K = 16$ in dynamic feedforward layer ($K = 16$), (5) remove the style discriminator D_{style}^{δ} (**w/o D_{style}^{δ}**), (6) remove triplet loss (**w/o L_{trip}^{δ}**), (7) remove the lip-sync discriminator D_{sync} (**w/o D_{sync}**), and (8) our full model (**Full**). The results are shown in Table IV and Fig. 8.

Since all variants utilize the same image renderer, they obtain similar SSIM and CPBD scores. Compared to **Full**, both **w/o DyFFN** and **w/o StyQ** achieve lower scores in F-LMD, M-LMD, and Sync_{conf}. Results in Fig. 8 reveal that without using the style code as a query (**w/o StyQ**), stable facial expressions and good lip-sync can still be realized; however, the consistency between the generated and reference expressions decreases significantly. On the other hand, While **w/o DyFFN** generally produces animations that maintain consistent expressions with the reference style clip, it sometimes leads to unstable facial animations (see Fig. 8). Therefore, using the style code as a query enhances the consistency between the synthesized and reference expressions, while using the adaptive feed-forward layer improves the stability of synthesized expressions and the accuracy of the mouth shape under various styles.

We empirically observe that $K = 8$ is the optimal setting for our task. Without D_{style}^{δ} and L_{trip}^{δ} , the F-LMD and M-LMD scores drop dramatically. This implies that the style discriminator and the triplet constraint compel our framework to better perceive the stylized facial motion patterns. Furthermore, the results show poor lip synchronization when D_{sync} supervision is not included. Fig. 8 and our demo video more clearly demonstrate the improvement each component brings to the final results.

D. Style Space Inspection

1) Style Space Visualization: For ease of visualization, We project the style codes to a 2D space using t-distributed stochastic neighbor embedding (t-SNE) [67].

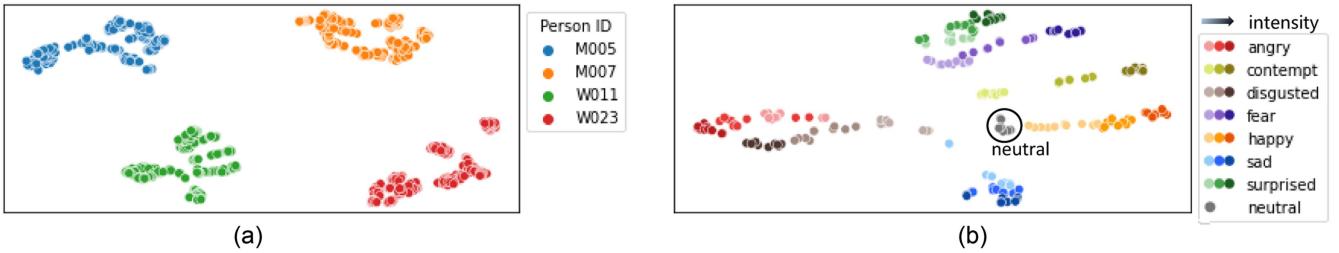


Fig. 10. (a) Visualization of the expression style codes of four speakers in MEAD. (b) Visualization of the emotional expression style codes of the speaker W011 in MEAD. Darker colors indicate higher emotion intensity.

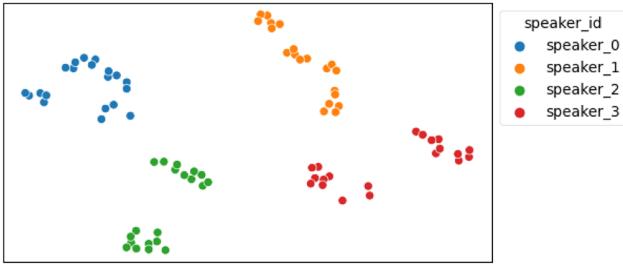


Fig. 11. Visualization of the head pose style codes of four speakers in HeadMotion dataset.

To visualize the expression style codes, we selected four speakers from the MEAD dataset. Each speaker had 22 expression styles (7 emotions x 3 levels plus one neutral style). For each style, we randomly selected 10 video clips to extract style codes. In Fig. 10(a), each speaker is marked with a distinct color. As shown, the style codes of the same speaker cluster in the style space. This implies that the expression styles of one speaker are more similar to each other than to those of the same emotion from other speakers. Fig. 10(b) shows the style codes from one speaker in the MEAD dataset. Each style code is marked with a color corresponding to its emotion and intensity. Each group of style codes with the same emotion gathers into one cluster. In each cluster, the style codes of emotions with low intensity are close to those of the neutral emotion. Notably, some emotions show similar facial motion patterns, such as anger vs disgust and surprise vs fear. Thus, their style codes are close in the style space.

We selected 4 video clips (30–60 seconds) of 4 speakers from the HeadMotion dataset and randomly cropped 20 clips of 5–10 seconds from each video clip to extract the head pose style codes. In Fig. 11, we can see that the style codes of the same speaker lie in nearby space, demonstrating that our model can accurately capture individual head pose styles.

The aforementioned observations prove that our model is able to learn a semantically meaningful style space.

2) *Style Manipulation*: Thanks to the meaningful style space, our method can edit the speaking styles by manipulating style codes. As shown in Figs. 12 and 13, when linearly interpolating between two style codes extracted from unseen style clips, the speaking styles of generated videos transition smoothly. Through interpolation, our method is able to control the style



Fig. 12. Interpolation results between 2 expression styles. For each interpolated style, we show the results of 2 reference speakers.



Fig. 13. Interpolation results between 2 head pose styles. For each style, we generate a video, and each video showcases two frames as examples.

intensity (by interpolating the style with a neutral style) and create new speaking styles.

E. User Study

We conduct two groups of user studies involving 36 participants. In the first group, we ask participants to rate the visual effects of generated videos using various methods, including Wav2Lip, PC-AVS, AVCT, EAMM, GC-AVT, ground truth, and our method. We generate three videos for each method and use the ground truth video as the expression style reference. Participants rate each video on a scale of 1–5 for lip sync quality, realness of results, and expression style consistency between the generated videos and the style reference. The mean scores are reported in Table V. Our method outperforms existing methods in all aspects, particularly in style consistency.

In the second group, we ask participants to rate the head motions in videos generated by MakeItTalk, Audio2Head, ground truth, and our method. We also generate three videos for each method and we use the ground truth video as the head pose style reference. Participants rated each video on a scale of 1–5 for the naturalness of head motion, the synchronization between head pose and audio rhythm (head motion sync), and the consistency

TABLE V
RESULTS OF THE USER STUDY ON VISUAL EFFECTS

Method	Wav2Lip	PC-AVS	AVCT	EAMM	GC-AVT	Ground Truth	Ours
Lip Sync	3.45	3.23	3.24	1.89	3.41	4.47	3.52
Video Realness	1.67	2.03	2.74	1.39	1.43	4.24	3.06
Expression style Consistency	1.22	1.68	1.62	1.80	2.65	1.85	3.46

The scores range from 1 to 5. Large scores indicate better perception. Here, the average scores across 24 videos are reported. All video are generated using the same input.

TABLE VI
RESULTS OF THE USER STUDY ON THE GENERATED HEAD MOVEMENTS

Method	MakeitTalk	Audio2Head	Ground Truth	Ours
Head motion naturalness	1.66	3.41	4.91	3.52
Head motion sync	2.12	3.94	4.86	3.78
Head pose style Consistency	1.16	2.17	4.70	3.64



Fig. 14. Results with the personalized sad expression styles. The style reference videos are collected from internet sources. For more details, please zoom in or watch our demo video.

of head pose style between the generated videos and the style reference. The mean scores are reported in Table VI. MakeitTalk receives low scores on all questions, indicating that it fails to produce natural-looking head movements. Although Audio2Head achieves competitive scores with our method on head motion naturalness and head motion sync, it is unable to control the style of the generated movements, thus failing to create diverse head motion sequences. These comparisons further validate the superiority of our approach.

F. Discussion

1) *Further Analysis:* Fig. 14 displays the results obtained using personalized sad expression styles. The figure illustrates that our method can capture subtle differences in the motion patterns of personalized expression styles and showcase them in the generated videos. Additionally, our method successfully learns the spatio-temporal representations of previously unseen speaking styles, which is different from methods such as GC-AVT that only transfer the static expressions of reference videos. Furthermore, Fig. 15 displays additional results produced by our method, demonstrating our method's capability of animating talking head videos for non-photorealistic paintings. This showcases the promising generation ability of our method beyond face photography.

Our method has the capability to generate expressive talking videos in real-time by utilizing just an audio clip, a reference



Fig. 15. More results generated by our method. Each video is generated with a random expression style and a random head pose style. In this figure, we display one frame from each video. The top row shows the corresponding reference speakers.

speaker image, and alternative style reference video clips. This technology has the ability to generalize to unseen style clips and various types of facial photography. This feature opens up many interesting practical applications, including the creation of short videos and visual dubbing. This technology can be particularly useful in the entertainment industry, where it can be used to dub a foreign language film or TV show into another language while maintaining the same emotional intensity and facial expressions. Additionally, this technology can be used to create personalized videos for individuals, such as a special greeting from a celebrity or a personalized message from a loved one.

2) *Limitation and Future Work:* While our method produces convincing results, there are still some limitations. First, our method fails to extract reasonable expression styles from style reference videos with extreme head poses and side views. Additionally, in extreme expressions, our method doesn't always fully close the lips for some phonemes, such as *p*, *b*, and *m*. Second, our approach is also restricted by the length of the

reference video. We find that when the length of the expression reference video is less than 0.5 seconds, or the length of the head pose reference video is less than 3 seconds, the network has difficulty extracting the appropriate style. Furthermore, if there is a significant style difference within the same reference video, our method yields uncertain results.

Since we completely remove the articulation-irrelevant information in the stylized expression generation branch, including the audio rhythm, there is a possibility that the facial expression rhythm in the synthetic video does not match the audio. In our future work, we plan to disentangle the audio rhythm information and integrate it into our framework.

Furthermore, limited by the image renderer, our methods may produce artifacts around the mouth when generating intense facial expressions, and around the head when generating large head movements. In our future work, we aim to expand the emotional talking video datasets and develop more advanced rendering techniques.

VI. CONCLUSION

In this paper, we introduce a new framework called *StyleTalk++*, which generates one-shot audio-driven talking faces with diverse speaking styles. Our method effectively extracts expression styles and head pose styles from arbitrary style reference videos, and then injects them into the generated talking face videos using a unified style-controllable framework. In contrast to previous works, our approach captures the spatio-temporal co-activations of speaking styles from the style reference videos, leading to authentic stylized talking face videos. Extensive experiments show that our method produces photo-realistic talking head videos with conditional speaking styles while achieving more accurate lip-sync and better identity preservation than the state-of-the-art.

ACKNOWLEDGMENTS

The authors would like to express our gratitude to Xinya Ji, Borong Liang, and Yan Pan for generously assisting us with the comparisons. We also thank Ran Yi, Zipeng Ye, and Zhiyao Sun for sharing their HeadMotion dataset with us. Additionally, we would like to thank Lincheng Li and Zhimeng Zhang for their valuable contributions to our discussions. Finally, we would like to thank Fuxi Youling Crowdsourcing for their support of our user research.

REFERENCES

- [1] S. Wang, L. Li, Y. Ding, C. Fan, and X. Yu, “Audio2head: Audio-driven one-shot talking-head generation with natural head motion,” in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1098–1105.
- [2] S. Wang, L. Li, Y. Ding, and X. Yu, “One-shot talking face generation from single-speaker audio-visual correlation learning,” in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 2531–2539.
- [3] Y. Ma et al., “StyleTalk: One-shot talking head generation with controllable speaking styles,” in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 1896–1904.
- [4] K. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. Jawahar, “A lip sync expert is all you need for speech to lip generation in the wild,” in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 484–492.
- [5] H. Zhou, Y. Sun, W. Wu, C. C. Loy, X. Wang, and Z. Liu, “Pose-controllable talking face generation by implicitly modularized audio-visual representation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4176–4186.
- [6] Z. Zhang, L. Li, Y. Ding, and C. Fan, “Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3661–3670.
- [7] F. Yin et al., “StyleHeat: One-shot high-resolution editable talking face generation via pretrained StyleGAN,” 2022, *arXiv:2203.04036*.
- [8] K. Wang et al., “MEAD: A large-scale audio-visual dataset for emotional talking-face generation,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 700–717.
- [9] S. Sinha, S. Biswas, R. Yadav, and B. Bhowmick, “Emotion-controllable generalized talking face generation,” 2022, *arXiv:2205.01155*.
- [10] X. Ji et al., “EAMM: One-shot emotional talking face via audio-based emotion-aware motion model,” 2022, *arXiv:2205.15278*.
- [11] B. Liang et al., “Expressive talking head generation with granular audio-visual control,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3387–3396.
- [12] P. Safari, M. India, and J. Hernando, “Self-attention encoding and pooling for speaker recognition,” 2020, *arXiv: 2008.01077*.
- [13] H. Zhu, M.-D. Luo, R. Wang, A.-H. Zheng, and R. He, “Deep audio-visual learning: A survey,” *Int. J. Automat. Comput.*, vol. 18, pp. 351–376, 2021.
- [14] L. Chen, G. Cui, Z. Kou, H. Zheng, and C. Xu, “What comprises a good talking-head video generation?: A survey and benchmark,” 2020, *arXiv: 2005.03201*.
- [15] S. Suwanjanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, “Synthesizing Obama: Learning lip sync from audio,” *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, 2017.
- [16] O. Fried et al., “Text-based editing of talking-head video,” *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–14, 2019.
- [17] J. Thies, M. Elgarib, A. Tewari, C. Theobalt, and M. Nießner, “Neural voice puppetry: Audio-driven facial reenactment,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 716–731.
- [18] L. Li et al., “Write-a-speaker: Text-based emotional and rhythmic talking-head generation,” in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2021, pp. 1911–1920.
- [19] C. Zhang et al., “3D talking face with personalized pose dynamics,” *IEEE Trans. Vis. Comput. Graph.*, vol. 29, no. 2, pp. 1438–1449, Feb. 2023.
- [20] L. Song, W. Wu, C. Qian, R. He, and C. C. Loy, “Everybody’s talkin’: Let me talk as you want,” 2020, *arXiv: 2001.05201*.
- [21] X. Ji et al., “Audio-driven emotional video portraits,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14 080–14 089.
- [22] A. Lahiri, V. Kwatra, C. Frueh, J. Lewis, and C. Bregler, “Lipsync3D: Data-efficient learning of personalized 3D talking faces from video using pose and lighting normalization,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2755–2764.
- [23] C. Zhang et al., “Facial: Synthesizing dynamic talking face with implicit attribute learning,” in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 3867–3876.
- [24] Y. Guo, K. Chen, S. Liang, Y. Liu, H. Bao, and J. Zhang, “AD-NeRF: Audio driven neural radiance fields for talking head synthesis,” 2021, *arXiv:2103.11078*.
- [25] X. Liu, Y. Xu, Q. Wu, H. Zhou, W. Wu, and B. Zhou, “Semantic-aware implicit neural audio-driven video portrait generation,” 2022, *arXiv:2201.07786*.
- [26] J. S. Chung, A. Jamaludin, and A. Zisserman, “You said that?” 2017, *arXiv: 1705.02966*.
- [27] Y. Song, J. Zhu, D. Li, X. Wang, and H. Qi, “Talking face generation by conditional recurrent adversarial network,” 2018, *arXiv: 1804.04786*.
- [28] L. Chen, Z. Li, R. K. Maddox, Z. Duan, and C. Xu, “Lip movements generation at a glance,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 520–535.
- [29] K. Vougioukas, S. Petridis, and M. Pantic, “Realistic speech-driven facial animation with GANs,” *Int. J. Comput. Vis.*, vol. 128, pp. 1398–1413, 2020.
- [30] D. Das, S. Biswas, S. Sinha, and B. Bhowmick, “Speech-driven facial animation using cascaded GANs for learning of motion and texture,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 408–424.
- [31] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang, “Talking face generation by adversarially disentangled audio-visual representation,” in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2019, pp. 9299–9306.
- [32] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, “Hierarchical cross-modal talking face generation with dynamic pixel-wise loss,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7832–7841.

- [33] O. Wiles, A. Koepke, and A. Zisserman, "X2Face: A network for controlling face generation using images, audio, and pose codes," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 670–686.
- [34] L. Chen et al., "Talking-head generation with rhythmic head motion," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 35–51.
- [35] Y. Zhou, X. Han, E. Shechtman, J. Echevarria, E. Kalogerakis, and D. Li, "Makelittalk: Speaker-aware talking-head animation," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–15, 2020.
- [36] N. Sadoughi and C. Busso, "Speech-driven expressive talking lips with conditional sequential generative adversarial networks," *IEEE Trans. Affect. Comput.*, vol. 12, no. 4, pp. 1031–1044, Fourth Quarter 2021.
- [37] H. Wu, J. Jia, H. Wang, Y. Dou, C. Duan, and Q. Deng, "Imitating arbitrary talking style for realistic audio-driven talking face synthesis," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 1478–1486.
- [38] S. Gururani, A. Mallya, T.-C. Wang, R. Valle, and M.-Y. Liu, "Space: Speech-driven portrait animation with controllable expression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 20 914–20 923.
- [39] Y. Lu, J. Chai, and X. Cao, "Live speech portraits: Real-time photorealistic talking-head animation," *ACM Trans. Graph.*, vol. 40, no. 6, pp. 1–17, 2021.
- [40] R. Yi, Z. Ye, J. Zhang, H. Bao, and Y.-J. Liu, "Audio-driven talking face video generation with learning-based personalized head pose," 2020, *arXiv: 2002.10137*.
- [41] Y. Deng, J. Yang, S. Xu, D. Chen, Y. Jia, and X. Tong, "Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshop*, 2019, pp. 285–295.
- [42] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D face model for pose and illumination invariant face recognition," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Survell.*, 2009, pp. 296–301.
- [43] Y. Guo, J. Cai, B. Jiang, J. Zheng, and J. Zhang, "CNN-based real-time dense face reconstruction with inverse-rendered photo-realistic face images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1294–1307, Jun. 2019.
- [44] X. Dong and J. Shen, "Triplet loss in siamese network for object tracking," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 459–474.
- [45] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," 2019, *arXiv: 1901.02860*.
- [46] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [47] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [48] I. Goodfellow et al., "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 2672–2680, 2014.
- [49] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1125–1134.
- [50] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.
- [51] X. Yu and F. Porikli, "Face hallucination with tiny unaligned images by transformative discriminative neural networks," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2017, pp. 4327–4333.
- [52] X. Yu and F. Porikli, "Hallucinating very low-resolution unaligned and noisy face images by transformative discriminative autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3760–3768.
- [53] X. Yu, B. Fernando, B. Ghanem, F. Porikli, and R. Hartley, "Face super-resolution guided by facial component heatmaps," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 217–233.
- [54] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2794–2802.
- [55] B. Yang, G. Bender, Q. V. Le, and J. Ngiam, "CondConv: Conditionally parameterized convolutions for efficient inference," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 1307–1318, 2019.
- [56] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of styleGAN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8110–8119.
- [57] W. Wang and Z. Tu, "Rethinking the value of transformer components," in *Proc. 28th Int. Conf. Comput. Linguistics*, 2020, pp. 6019–6029.
- [58] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.
- [59] Y. Ren, G. Li, Y. Chen, T. H. Li, and S. Liu, "Pirenderer: Controllable portrait image generation via semantic neural rendering," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 13 759–13 768.
- [60] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A large-scale speaker identification dataset," 2017, *arXiv: 1706.08612*.
- [61] R. Yi et al., "Predicting personalized head movement from short video and speech signal," *IEEE Trans. Multimedia*, vol. 25, pp. 6315–6328, 2023.
- [62] A. Siarohin, S. Lathuilière, S. Tulyakov, E. Ricci, and N. Sebe, "First order motion model for image animation," *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 7137–7147, 2019.
- [63] A. Paszke et al., "Automatic differentiation in PyTorch," *NeurIPS W*, 2017.
- [64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv: 1412.6980*.
- [65] J. S. Chung and A. Zisserman, "Out of time: Automated lip sync in the wild," in *Proc. Asian Conf. Comput. Vis.*, Springer, 2016, pp. 251–263.
- [66] N. D. Narvekar and L. J. Karam, "A no-reference perceptual image sharpness metric based on a cumulative probability of blur detection," in *Proc. IEEE Int. Workshop Qual. Multimedia Experience*, 2009, pp. 87–91.
- [67] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.



Suzhen Wang received the BE degree from Southeast University, Nanjing, China, in 2016, and the MS degree from Zhejiang University, Hangzhou, China in 2019. He is currently an artificial intelligence researcher with Netease Fuxi AI Lab, Hangzhou. His research interests include computer vision, multimodal learning, image and video processing, animation generation and their application in games.



Yifeng Ma received the BEng degree from the Harbin Institute of Technology, China, in 2018. He is currently working toward the PhD degree with Department of Computer Science and Technology, Tsinghua University. His research interest focuses on computer vision and machine intelligence.



Yu Ding (Member, IEEE) received the PhD degree in computer science from Telecom ParisTech in Paris, France. He is currently an artificial intelligence expert, leading the virtual human team with Netease Fuxi AI Lab, Hangzhou, China. His research interests include virtual human, deep learning, image and video processing, talking-head generation, animation generation, multimodal computing, affective computing, and embodied conversational agent.



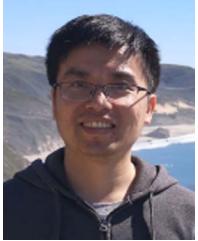
Zhipeng Hu received the graduation degree from Zhejiang University and once worked with MSRA. He vice president of NetEase Group. His major interest includes Game Design, Game AI, Computer Vision and Graphics. He is also the co-founder of NetEase Fuxi AI Lab and has great insight into the cross-domain of game and AI. He has been responsible for developing several most successful and popular games in worldwide, such as Ghost and Justice.



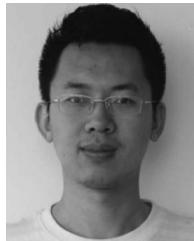
Changjie Fan received the doctor's degree in computer science from the University of Science and Technology of China. He is the director of NetEase Fuxi AI Lab. His research interest is in machine learning, including multiagent systems, deep reinforcement learning, game theory, and knowledge discovery.



Zhidong Deng (Member, IEEE) received the BS degree in computer science and automation from Sichuan University, Chengdu, China, in 1986 and the PhD degree in computer science and automation from the Harbin Institute of Technology, Harbin, China, in 1991. He is a professor with the Department of Computer Science and Technology, Tsinghua University, Beijing, China. He has been a full professor with Tsinghua University since 2000. His current research areas include artificial intelligence, computational neuroscience, autonomous driving, and advanced robotics.



Tangjie Lv received the PhD degree in computational mathematics from Peking University. He is the head of the AI team of NetEase Fuxi Lab. After graduation, he worked for NetEase Games and Tencent AI Lab. Presently, he is responsible for the AI research and technology development in games and other entertainment scenes. His research interests include reinforcement learning, machine learning and game AI. He has led his team to achieve successful AI implementation in several NetEase games.



Xin Yu received the BS degree in electronic engineering from the University of Electronic Science and Technology of China, Chengdu, China, in 2009, the PhD degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2015, and the PhD degree from the College of Engineering and Computer Science, Australian National University, Canberra, Australia, in 2019. He is currently a senior lecturer with the University of Queensland. His research interests include computer vision and image processing.