Personalized Audio-Driven 3D Facial Animation via Style-Content Disentanglement

Yujin Chai, Tianjia Shao, Yanlin Weng, and Kun Zhou, *Fellow, IEEE*

Abstract—We present a learning-based approach for generating 3D facial animations with the motion style of a specific subject from arbitrary audio inputs. The subject style is learned from a video clip (1-2 minutes) either downloaded from the Internet or captured through an ordinary camera. Traditional methods often require many hours of the subject's video to learn a robust audio-driven model and are thus unsuitable for this task. Recent research efforts aim to train a model from video collections of a few subjects but ignore the discrimination between the subject style and underlying speech content within facial motions, leading to inaccurate style or articulation. To solve the problem, we propose a novel framework that disentangles subject-specific style and speech content from facial motions. The disentanglement is enabled by two novel training mechanisms. One is two-pass style swapping between two random subjects, and the other is joint training of the decomposition network and audio-to-motion network with a shared decoder. After training, the disentangled style is combined with arbitrary audio inputs to generate stylized audio-driven 3D facial animations. Compared with start-of-the-art methods, our approach achieves better results qualitatively and quantitatively, especially in difficult cases like bilabial plosive and bilabial nasal phonemes.

Index Terms—Audio-driven animation, facial animation, style learning, style-content disentanglement, facial motion decomposition

1 Introduction

Audio-driven 3D facial animation has been widely studied due to its benefits to various applications such as 3D game animation, 3D teleconference, and virtual reality. Especially, *personalized* audio-driven facial animation has attracted great attention in recent years since personal motion styles are ubiquitous in people talking. For example, while all people have to press their lips to pronounce the phonemes /b/, /p/, and /m/, some people prefer to protrude lips at the same time. The indispensable lip pressing action *shared* across all people reflects the underlying *speech content*, and the *person-specific* protruding action shows the *facial motion style*. Such person-specific styles are crucial for high-fidelity facial animations. In this paper, we aim to generate personalized audio-driven 3D facial animations for an ordinary user. She/he only needs to take a short selfie video or download an Internet video clip (1-2 minutes) as training data, and the subject-specific facial motion style can be learned along with the reconstructed 3D motions. Afterward, given an arbitrary audio clip as input, we can produce a personalized 3D facial animation for the target subject.

• *The authors are with the State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou, Zhejiang 310058, China. E-mail: {11821001, tjshao}@zju.edu.cn, weng@cad.zju.edu.cn, kunzhou@acm.org.*

Manuscript received 29 June 2022; revised 9 October 2022; accepted 26 November 2022. Date of publication 19 December 2022; date of current version 30 January 2024.
This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFF0902302, in part by NSF China under Grant 62172357, and in part by XPLORER PRIZE.
(Corresponding author: Yanlin Weng.)
Recommended for acceptance by H. Huang.
This article has supplementary downloadable material available at https://doi.org/10.1109/TVCG.2022.3230541, provided by the authors.
Digital Object Identifier no. 10.1109/TVCG.2022.3230541

Traditional methods have difficulties to be applied for this task. They commonly require sufficient training data to cover enough speech content for learning a robust model, for instance, tailored corpus [1] or enormous video data for a specific subject [2], [3]. However, in our scenario, the given video clip only has limited coverage of speech content. Some recent works [4], [5], [6] incorporate auxiliary data from many other subjects, e.g., public talking face datasets, to improve the content coverage. Whereas, the involvement of other subjects introduces many motion styles different from that of the target subject. These methods tried different ways to distinguish the target subject's style from other styles, including style mapping matrices [4], model fine-tuning [5], and handcrafted style codes [6]. While these works can generate faithful results under the target subject's style, they may still fail to pronounce a word properly in some cases, especially at the bilabial plosive phonemes /b/, /p/ and the bilabial nasal phoneme /m/ (see an example in Figs. 8 and 9). We make a key observation that it is because these methods lack a mechanism to discriminate the subject-specific *motion style* and underlying *speech content*. In the aforementioned difficult cases (/b/, /p/, and /m/), the indispensable lip pressing action is the content. Without clear disentanglement of style and content, whereas, previous methods may wrongly treat the content as a kind of style. Consequently, their models may omit the necessity of lip pressing and fail to pronounce these phonemes properly.

To tackle the above problem, we propose a novel approach to explicitly disentangle the subject-specific motion style and the underlying speech content within facial motions during network training. In this way, we can let the model not only capture the target style well but also achieve more accurate lip motions. Essentially, we design a novel decomposition network with two encoders to decompose the 3D facial motions reconstructed from a video clip

into the style and content latent codes. The disentangled style code is naturally utilized to distinguish the target style from others and can be combined with arbitrary speech audio clips to generate personalized audio-driven 3D facial animations. The core of our method is two novel training mechanisms, which impose explicit supervision to ensure the clean disentanglement of style and content. First, we propose a novel two-pass style swapping mechanism. In the first pass, we swap the style codes between two subjects and force the decoded new motions to preserve the original speech content but with each other's style. In the second pass, we swap back the styles and compel the decoded motions to be identical to the inputs. In this way, we let the network explicitly know the respective roles the style and content codes should play, rather than mingling them without any supervision. Second, we jointly train the decomposition network and an audio-to-motion network as a unified framework. The audio-to-motion network has an audio encoder and shares the decoder with the decomposition network. Since audio features only contain speech content information, we can force the decomposed content codes to be close to the encoded audio features by jointly training the two networks with the same ground truth. In this way, the content codes are further disentangled from the style information. The audio-to-motion network will also be used to generate new animations from the audio input after training.

Technically, we reconstruct the 3D facial animation from the input video by the FLAME [7] and use the reconstructed data together with an auxiliary 4D facial dataset, i.e., VOCASET [8], as training data. All 3D face meshes are aligned to the corresponding identity template shapes to remove head poses. The facial motions are represented as vertex displacements by subtracting the identity template shapes after alignment. The audio signals are preprocessed by the pre-trained automatic speech recognition (ASR) network DeepSpeech [9] to minimize the effect of voice pitches, recording noise, and accents. When jointly training the two networks, we input paired facial motions and audio features into the decomposition and audio-to-motion networks. The shared decoder takes as input the style code along with the content codes or encoded audio features to generate 3D facial motions. The input facial motions are also used as ground truth to supervise the decoder outputs. The aforementioned two-pass style swapping strategy is applied to two data in a randomly sampled training mini-batch. After training, the disentangled style code for the target subject is obtained. During the generation, it is sent to the audio-to-motion network along with arbitrary speech audio to produce personalized 3D facial animations.

Extensive qualitative and quantitative experiments have shown that our approach can robustly learn the subject-specific motion style from a short selfie video or an Internet video clip, and generate high-quality personalized audio-driven 3D facial animations. To assess the quality of our results, please watch the supplementary video[1].

In summary, the main contributions of our approach include:

- We propose a novel decomposition network to explicitly disentangle the subject-specific motion style and underlying speech content from 3D facial motions. The disentanglement enables more accurate style and articulation. To our best knowledge, our work is the first deep-learning attempt for the disentanglement of subject-specific style and speech content from 3D facial motions.
- We propose two novel training mechanisms tailored to our framework, including two-pass style swapping and joint training with decoder sharing.
- We compare our results with state-of-the-art methods, and ours outperform others in both qualitative and quantitative comparison.

2 Related Works

Since our main task is to learn personalized audio-driven 3D facial animation, we mainly investigate previous works about style learning in the audio-driven facial animation field.

2.1 Non-Deep-Learning Methods

Deng et al. [10], [11] acquire emotional signals by subtracting aligned neutral speech motion data from emotional speech motion data and reduce dimensions by Principal Component Analysis (PCA). In the generation phase, they generate an emotional motion curve by patch sampling and add it to a neutral motion curve to obtain the final output. Cao et al. [12] apply Independent Component Analysis (ICA) to decompose facial motions, represented by landmarks, into content and emotional parts. Smooth mappings between different emotions are learned through a Radial Basis Function (RBF) approximation. In the generation phase, they search for the best path in a directed graph pre-built on training data to generate an intermediate sequence. The content and emotional parts in each intermediate frame are decomposed by ICA, and the emotional part is mapped to the target emotion by RBF. Finally, they reassemble the content and the mapped emotional components by inverse ICA to generate the final output. Edwards et al. [13] generate a facial animation from phonemes according to pre-defined linguistics rules in procedure. Meanwhile, they propose jaw-lip (JALI) based 2-dimensional parameters to enable easy manual control of talking styles for artists. These early methods mainly focus on emotion styles, different from our target of learning subject-specific motion styles.

2.2 Deep-Learning Methods

2.2.1 *Subject-Specific Models*

Many works [1], [2], [3], [14], [15], [16] train a deep neural network for one specific person. Taylor et al. [3] train a network on the KB-2k dataset, containing 2543 different sentences from one speaker, to predict Active Appearance Model (AAM) parameters from phonetic features. Suwajanakorn et al. [2] synthesize high-quality talking videos for Barack Obama based on million frames from Obama's weekly addresses. Karras et al. [1] train a specific network for each subject and control the emotional state of output animations. Their method requires expensive 4D facial scans of

1. Also available at: https://chaiyujin.github.io/psfa

each subject reading carefully tailored corpus. Given a target video clip of a talking face, Song et al. [14] and Wen et al. [15] both train an audio-to-expression network to predict 3D Morphable Model (3DMM) coefficients and a neural video rendering network to convert predicted 3DMM animations into photo-realistic videos. Similarly, Zhang et al. [16] also use 3DMM as an intermediate representation and train two networks, i.e., audio-to-animation and animation-to-video, to predict 3DMM animations from audio and synthesize videos guided by the flow computed from 3DMM predictions. These methods either require a tailored corpus or a large amount of training data for a specific speaker. Otherwise, they may suffer from poor generalization when training data only have limited coverage of speech content.

2.2.2 Explicit Style Learning

As it is not always possible to capture lots of data for one subject, some recent works incorporate data from many different subjects to enlarge speech content coverage and improve speech robustness. Since the motion styles from different data vary from each other, to generate animations under the target style, these works try to control the output style explicitly. Cudeiro et al. [8] propose the VOCA model to generate detailed 3D facial animation synchronized with audio input and leverage one-hot encoding vectors to control different styles learned from training subjects. This strategy is also applied in the following works [17], [18]. Thies et al. [4] train a shared audio-to-expression network among all training subjects. Additionally, a specific matrix mapping expression latent codes to personalized 3DMM coefficients and a neural rendering network for each subject are trained with the shared network in an end-to-end way. To mimic a new style from a video after training, the shared audio-to-expression network keeps fixed, and a mapping matrix is inferred by the data reconstructed from the video, while the neural rendering network has to be trained from scratch. The style modeling ability of their method can be limited because they only use linear mapping matrices to handle different subject styles. Yi et al. [5] propose an LSTM-based audio-to-motion network to predict 3DMM coefficients and a Memory-Augmented GAN model for photo-realistic video generation. They pre-train the LSTM and GAN models on large-scale datasets and fine-tune them to fit a new style from the target subject's talking video. However, they do not distinguish different styles during pre-training. The audio-to-motion network may be confused when the same sentences are said in different styles, and the subsequent fine-tuning can also be affected. Wu et al. [6] propose a framework to imitate an arbitrary motion style from a reference video. They first predict stylized 3DMM coefficients from audio guided by the style code obtained from the reference video and then synthesize a photo-realistic video given a static portrait image. The crucial style codes, however, may have insufficient style information, because they are handcrafted based on the statistics of the reconstructed 3DMM coefficients from the reference video.

Another interesting effort is MeshTalk proposed by Richard et al. [19]. MeshTalk can work with or without a style reference 3D animation sequence. When a reference sequence is available, the expression of model output is controlled, mainly on the upper face area. This functionality goes in a different direction from our task to learn subject-specific motion styles. We focus on lower face movements which obviously play an important role in subject-specific styles, especially lip motions.

2.2.3 Implicit Methods

Shimba et al. [20] and Pham et al. [21], [22] do not explicitly control or predict affective states but can generate emotional facial talking animation by learning from data implicitly. VisemeNet proposed by Zhou et al. [23] first predicts phoneme group probabilities and landmark displacements from audio features and then predicts rig parameters from both the intermediate results and input audio features to drive JALI [13]. Their intermediate landmark displacements are proven useful for implicit control of the emotional style of outputs. These methods mainly focus on emotional states. Since the audio data contain emotional information, it is reasonable to learn emotional styles implicitly. In our task, however, we want to generate personalized animations from arbitrary speech audio even said by another person. It cannot be achieved in such an implicit way.

2.3 Disentanglement

Even though our work is the *first* deep-learning attempt to disentangle the subject-specific motion style and the underlying speech content from 3D facial motions, disentanglement has been applied widely in many fields.

In the unsupervised image-to-image translation task, Liu et al. [24] make an assumption that images from two different domains can be encoded onto and generated from the same shared-latent space. Lee et al. [25] extend this idea by embedding images onto two spaces: a shared domain-invariant content space and a domain-specific attribute space. Lee et al. [26] further extend this method into multi-domain translation. In the 3D portrait stylization task, Han et al. [27] disentangle the geometry and texture styles by handling geometry and texture in sequence through two stages. Besides, in the first stage, they capture the coarse geometric style by facial landmark translation, in which the disentanglement of shape content and artistic style is achieved. In the visual dubbing task, Kim et al. [28] disentangle the mouth motions of the source actor and transfer them to the target actor's style.

The idea of disentanglement is also explored by some previous works in other deep-learning-based talking face generation tasks. For example, Zhu et al. [29] design a dynamic attention block to disentangle lip-related and identity-related information from previous frames in the image space. Zhou et al. [30] decompose video data into speech content and the speaker's appearance in the image space, while Zhou et al. [31] drive the talking face with the speech content and identity information decomposed from the input signal in the audio space. Both works from Mittal et al. [32] and Ji et al. [33] decompose audio signals into speech content and emotional stuff. Yao et al. [34] decompose a face into different parts, such as lips, eyes, and so on, in image space. Different from these works, our approach attempts to disentangle subject-specific motion style and

underlying speech content information from 3D facial motions, which is a completely different task.

3 Preliminaries

In order to make the paper more self-contained, we first give a brief review of the face model and auxiliary dataset used in the paper.

FLAME is a statistical model, representing a head with $N = 5023$ vertices and $K = 4$ joints (jaw, neck, and eyeballs) using linear blend skinning (LBS) and pose-dependent corrective blendshapes [7]. With the shape $\vec{\beta} \in \mathbb{R}^{|\vec{\beta}|}$, pose $\vec{\theta} \in \mathbb{R}^{3K+3}$, and expression $\vec{\psi} \in \mathbb{R}^{|\vec{\psi}|}$ parameters, the model is defined as

$$M(\vec{\beta}, \vec{\theta}, \vec{\psi}) = W(T_P(\vec{\beta}, \vec{\theta}, \vec{\psi}), \mathbf{J}(\vec{\beta}), \vec{\theta}, \mathcal{W}), \tag{1}$$

where

$$T_P(\vec{\beta}, \vec{\theta}, \vec{\psi}) = \overline{\mathbf{T}} + B_S(\vec{\beta}; \mathcal{S}) + B_P(\vec{\theta}; \mathcal{P}) + B_E(\vec{\psi}; \mathcal{E}). \tag{2}$$

$\overline{\mathbf{T}} \in \mathbb{R}^{3N}$ is the mean "zero-pose" template. $B_S(\vec{\beta}; \mathcal{S})$, $B_P(\vec{\theta}; \mathcal{P})$, and $B_E(\vec{\psi}; \mathcal{E})$ denote a shape blendshape function, pose-dependent corrective blendshapes, and expression blendshapes, respectively. Some vertices in $T_P(\vec{\beta}, \vec{\theta}, \vec{\psi})$ will be further rotated by the blend skinning function $W(\cdot)$ around the joints, computed from the function $\mathbf{J}(\vec{\beta})$, with linear smoothing weights $\mathcal{W} \in \mathbb{R}^{K \times N}$. Please see [7] for details.

VOCASET is a collection of 4D facial scans paired with speech audio captured at 60 frames per second (FPS) [8]. In total, the dataset contains 480 sequences from 12 subjects, 40 English sentences (3 to 5 seconds) per person. The sentences were deliberately selected to maximize phonetic diversity. For the 40 sentences spoken by each subject, 5 sentences are shared across all subjects, 15 sentences are spoken by three to five subjects (50 unique sentences), and the remaining 20 ones are spoken by one or two subjects (200 unique sentences). For each subject, there is an expressionless zero-pose template mesh representing the identity geometry. All scans in 40 sentences are aligned with the template to remove the head poses. By subtracting the template mesh from all frames, we can use the vertex displacements to represent the facial motions. Please see [8] for details.

4 Method

4.1 Overview

Given a talking video clip (1-2 minutes) of a target subject, we aim to learn her/his facial motion style in 3D, so that we can generate personalized 3D facial animations robustly for the target subject from arbitrary speech audio. We first track the given video with the FLAME [7] to reconstruct 3D facial motions. Also, accompanying audio features are computed (Section 4.2). Considering the provided video only covers a limited amount of speech content, which is not enough for training a robust model, we enlarge the content coverage by utilizing auxiliary subjects' data from VOCASET [8]. To achieve more accurate style and articulation, we propose a unified framework, composed of an audio-to-motion network (Section 4.3) and a novel decomposition network (Section 4.4), to disentangle style and content information from 3D facial motions. The disentangled style information is naturally used to distinguish different subject styles. We propose two novel mechanisms to train our framework, including two-pass style swapping and joint training with decoder sharing, to supervise the disentanglement of style and content (Section 4.5). After training, the disentangled style code can be combined with unseen speech audio to generate personalized 3D facial animations for the target person (Section 4.6). The overall framework is depicted in Fig. 1.

4.2 Data Processing

4.2.1 3D Facial Motion Reconstruction

We basically follow DECA [35] to reconstruct 3D facial animations using FLAME, with several modifications made to suit our task. We mainly focus on the facial motions related to speech, so the step of detailed wrinkle reconstruction in the original work is skipped. We reconstruct the 3D facial animation in two steps. First, we randomly select a batch (32) of frames to solve FLAME parameters by sharing a single shape parameter $\vec{\beta}$ among the batch, so that each frame has exactly the same identity shape. Second, we fix the solved $\vec{\beta}$ and track the entire video to solve expression $\vec{\psi}$ and poses $\vec{\theta}$ frame by frame. Afterward, the pose parameters for head and eyeballs are ignored, as we do not deal with the head pose and eyeball movements. Only $\vec{\psi}$ and pose parameters for the jaw joint are kept. We also add an extra temporally smooth loss term to stabilize the motion reconstruction, whereas DECA focuses on single images.

The reconstructed identity mesh (according to $\vec{\beta}$) is denoted as $\mathbf{I} \in \mathbb{R}^{3N}$ and the mesh of the t-th frame is denoted as $\mathbf{v}_t \in \mathbb{R}^{3N}$. Similar to previous works [4], [5], [6], [8], [15], we remove the head movements and identity information by representing the facial motion as 3D vertex displacements from the identity shape after rigid alignment $\mathbf{y}_t = \mathbf{v}_t - \mathbf{I}$. A sequence of facial motions is $\mathbf{Y_i} = \{\mathbf{y}_t\}_{t \in \mathbf{i}}$, where $\mathbf{i}$ indicates contiguous frame indices in the sequence.

4.2.2 Audio Features

DeepSpeech [9] is an end-to-end deep neural network for Automatic Speech Recognition (ASR). It is a prevalent method for extracting audio features [4], [6], [8] due to its robustness to different audio sources, speakers, background noises, and accents. We follow previous works and use the open-source implementation [36] to extract audio features.

For the input audio, DeepSpeech predicts the probabilities of 29 characters at 50 frames per second, i.e., each frame lasts 20 milliseconds. We use the unnormalized log probabilities before Softmax normalization as features. Since short temporal audio information is proven critical for predicting natural lip motions due to co-articulation [37] and audiovisual asynchrony [38], we use a temporal window of DeepSpeech features lasting $20W$ milliseconds as the audio feature for each frame of facial motion, taking $10W$-milliseconds history and $10W$-milliseconds future audio context into account. Here W is the frame number of DeepSpeech features in one temporal window. We denote the audio feature paired with the t-th frame of facial motion as $\mathbf{x}_t \in \mathbb{R}^{W \times 29}$.

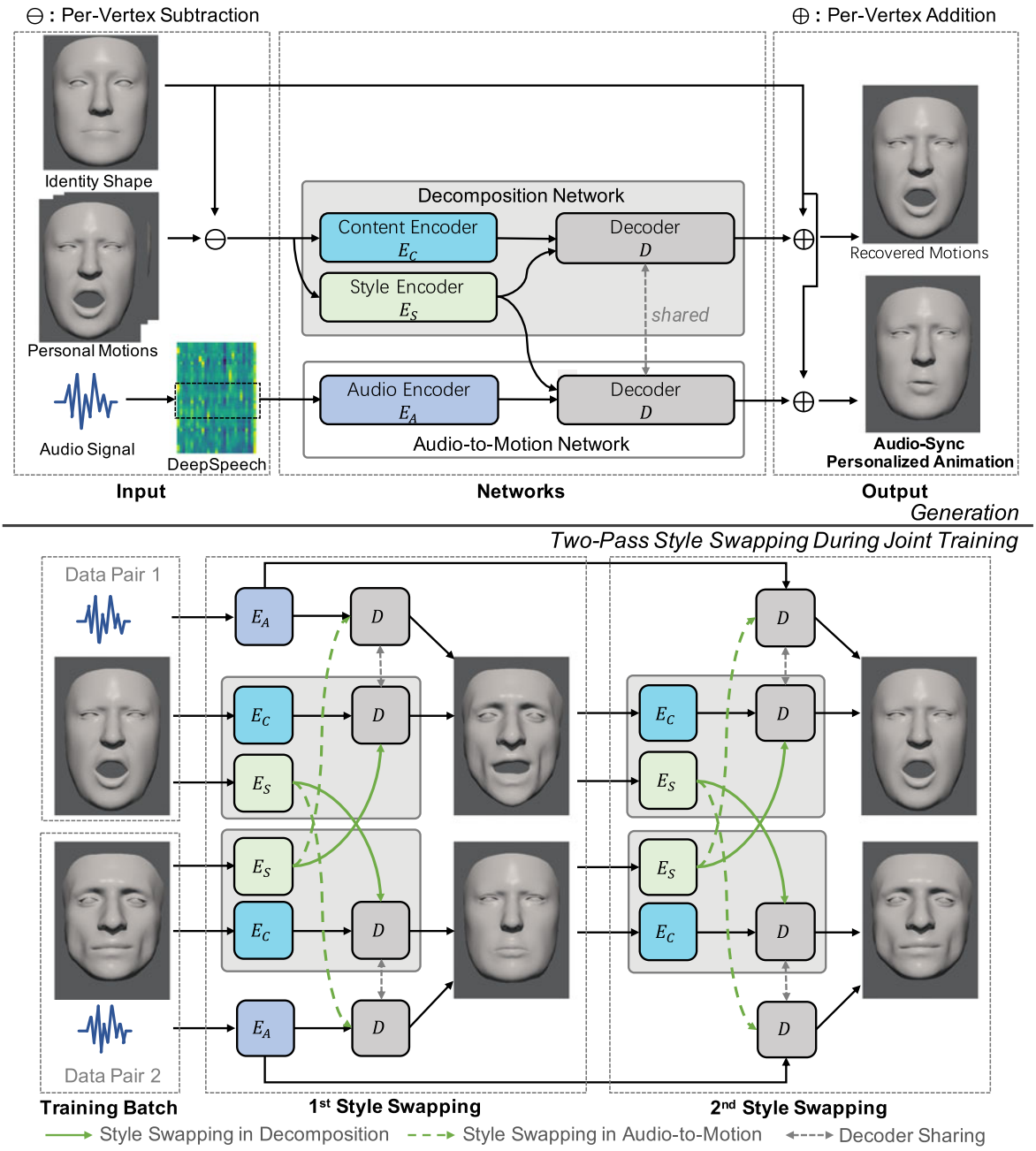


Fig. 1. Top: our framework contains two networks, a decomposition network disentangling the subject-specific motion style and underlying speech content from 3D facial motions, and an audio-to-motion network predicting audio-synchronized animation from audio features and style codes. Bottom: our novel training mechanisms include two-pass style swapping and joint training with decoder sharing, which force the networks to properly disentangle style and content. The identity shape subtraction and DeepSpeech features extraction are omitted in the bottom part for a clear visualization. All rendered meshes in the bottom part represent vertex displacements without identity shapes.

4.3 Audio-to-Motion Network

The audio-to-motion network, predicting vertex displacements synchronized with input audio features, is depicted in Fig. 2. Each input audio feature window is encoded by an audio encoder E_A to summarize the articulation and co-articulation context. Then a decoder D follows to decode the encoded audio features and predicts the vertex displacements under the corresponding subject-specific motion style controlled by a style code. Specifically, the decoder is composed of a casual sequential module and a displacement decoding module. The causal sequential module handles longer temporal dependency and smooths the audio latent codes temporally. Besides, the style code is injected into this module to explicitly control the style of final outputs. The displacement decoding module outputs vertex displacements frame by frame.

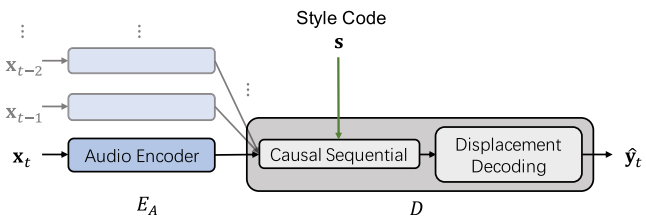


Fig. 2. Structure of the audio-to-motion network with style injection.

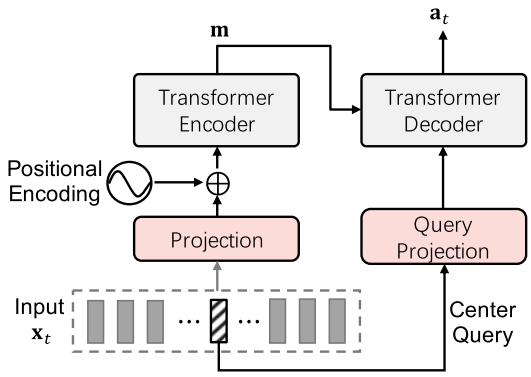


Fig. 3. Structure of the transformer-based audio encoder.

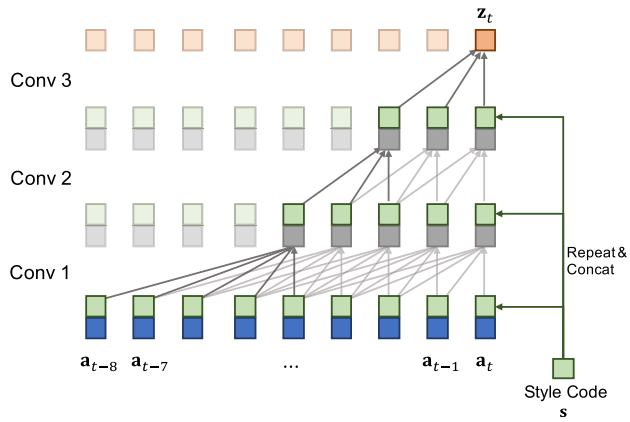


Fig. 4. Structure of the causal sequential module. A style code is repeated and concatenated with input feature maps before each convolution layer. Some nodes and links are dimmed just for better visualization.

4.3.1 Network Architecture

For the audio encoder, instead of using a fully convolutional encoder [1], [8] or a recurrent neural network [18], we adopt a transformer [17], [39] based architecture to summarize the features from a temporal window, which shows slightly better visual results in the early tests. Specifically, for the t-th frame, $\mathbf{x}_t$ is linearly projected and added with positional encoding, and then fed into a 3-layers transformer encoder to get memory $\mathbf{m} \in \mathbb{R}^{W \times d_{model}}$, where d_{model} is a parameter for the transformer. Since we want to summarize information from a temporal window, for the transformer decoder, we select the center single frame of $\mathbf{x}_t$ as input, which represents the exact audio feature aligned with the t-th frame of facial motion. It is linearly projected and fed into the transformer decoder to query over the memory $\mathbf{m}$, as shown in Fig. 3. The output is denoted as $\mathbf{a}_t \in \mathbb{R}^{1 \times d_{model}}$. As we use the center frame of $\mathbf{x}_t$ as a query, this transformer actually performs self-attention over $\mathbf{x}_t$ rather than the standard functionality of seq-to-seq mapping.

The causal sequential module is composed of three causal 1D convolution layers with a LeakyReLU activation function for each. It serves to deal with longer temporal dependency and smooth the outputs from the audio encoder. For the t-th frame, this module takes extra n history frames into account, namely $\{\mathbf{a}_{t-n}, \ldots, \mathbf{a}_{t-1}, \mathbf{a}_t\}$, so that temporal information longer than an audio feature window is covered. From another viewpoint, a 1D convolution layer can be seen as calculating a weighted sum of the inputs, so this convolutional architecture smooths inputs in nature. In addition, a style code is injected into this module to guide the facial motion style. We use the injection method where the style code s is repeated and concatenated with input feature maps before each convolution layer [1], [6], [8], [18]. The output is denoted as $\mathbf{z}_t$, and the architecture is depicted in Fig. 4.

The displacement decoding module is built with five fully connected layers. A LeakyReLU activation function is used after each layer except for the last two layers. The weights of the last layer are initialized with expression bases from FLAME and are trained as usual parameters. This module predicts vertex displacements $\hat{\mathbf{y}}_t$ from $\mathbf{z}_t$ for the t-th frame of animation.

4.3.2 A Quick Discussion of Style Code

The reconstructed data from the target subject's video are not enough to train an audio-to-motion network with good generalization capacity on unseen speeches due to the limited coverage of speech content. Even though DeepSpeech provides superior generalization to different audio signals, sufficient speech content coverage is still required to handle diverse phonetic combinations and complex co-articulation phenomena. However, it is not always possible to capture hours of data for the specific target subject. To get out of the awkward situation, we leverage auxiliary data to improve the speech generalization ability of the model. VOCASET is an ideal candidate due to its good phonetic coverage as described in Section 3.

However, the involvement of VOCASET inevitably brings the challenge of maintaining the target style, since the subjects from VOCASET can have very different motion styles from the target speaker's. Without the style code as an explicit control, the audio-to-motion network may totally obscure different styles. The reason why we inject the style code into the casual sequential module, rather than the displacement decoding module, is that motion styles are sometimes exhibited temporally. While the audio feature for each frame is a short temporal window, the injection in the casual sequential module allows the network to handle style in a longer temporal context if necessary.

To obtain an accurate style code, as the core module of our method, we propose a novel decomposition network to disentangle the style and content information from the target subject's 3D facial motions. The disentangled style latent code serves to explicitly control the motion style of the audio-to-motion network outputs.

4.4 Decomposition Network

Our proposed decomposition network has an encoder-decoder architecture. Two encoding branches encode the input sequence of vertex displacements into the style and content latent codes by the style encoder E_S and content encoder E_C respectively. A decoder D decodes the latent codes and recovers the input sequence then, as shown in Fig. 1.

For an input sequence $\mathbf{Y}_\mathbf{i}^p = \{\mathbf{y}_t^p\}_{t \in \mathbf{i}}$ with subject style p, two encoders extract the latent codes $\mathbf{s}_p$ (motion style) and $\mathbf{C}_\mathbf{i} = \{\mathbf{c}_t\}_{t \in \mathbf{i}}$ (speech content) respectively. As shown in Fig. 5, in each branch, a three-layers SpiralNet [40] is leveraged to extract information from graph-structured vertex

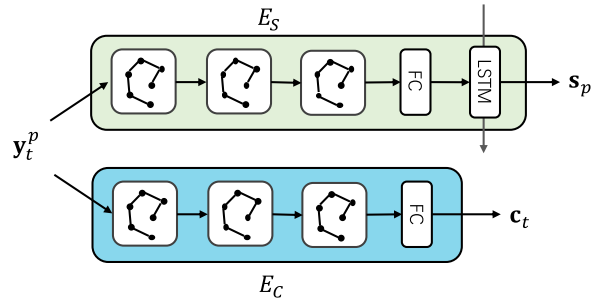


Fig. 5. Structure of the style encoder E_S and the content encoder E_C in the decomposition network.

displacements. The structure of SpiralNets in the two branches is identical, but the weights are not shared. A fully connected layer follows in both. For the branch of the style encoder, a standard single-directional LSTM layer is utilized to summarize the style over the entire sequence.

The decoder of the audio-to-motion network is reused in the decomposition network. Not only the structure is reused, but also the weights are shared. The decoder sharing is used together with a joint training strategy to ensure the content encoder only captures content information, which will be described later in Section 4.5.2.

4.5 Training

In this section, we explain the mechanisms and loss functions used to train our framework, so as to ensure the decomposition network is capable of valid style-content disentanglement.

4.5.1 Two-Pass Style Swapping

If we naively train the two networks together, a trivial solution may occur: the decoder simply underrates the style encoding branch and mainly relies on the encoded content information to fit the different facial motions. Consequently, the networks fail to learn the proper motion style of the target subject, which will be shown in Section 5.5.2. To overcome such a trivial solution, we apply two-pass style swapping over two data in a training mini-batch as shown in Fig. 6. In the first-pass style swapping, we swap the encoded styles between two data (green arrows in Fig. 6) and supervise the decoded facial motions to keep the original speech content but with a style from the other (orange double arrows in Fig. 6). In the second-pass style swapping, we swap the style latent codes encoded from the first-pass outputs again and supervise the decoded motions to be the same as the original inputs (red double arrows in Fig. 6). This mechanism compels the networks to clearly differentiate the roles played by style and content latent codes, rather than ignoring style or relying too much on content information. The loss function for this mechanism will be explained in Section 4.5.4.

4.5.2 Joint Training With Decoder Sharing

We apply a further constraint to force the content encoder E_C to capture the underlying speech content only rather than facial motion style by training the two networks jointly with sharing the decoder between them. During the training phase, we input paired facial motions and audio data into

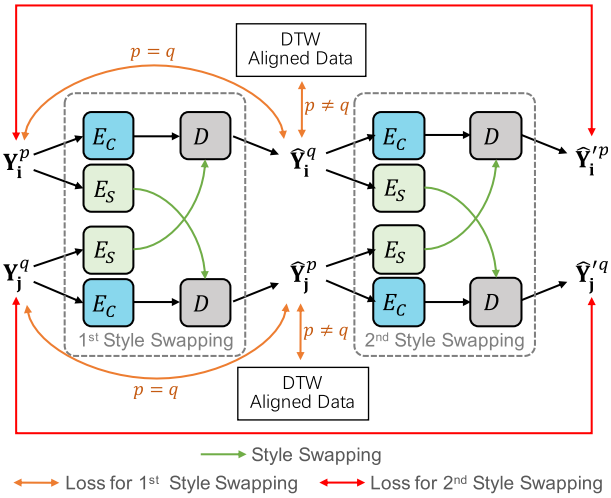


Fig. 6. The two-pass style swapping in the decomposition network. $\mathbf{Y}_\mathbf{i}^p$ and $\mathbf{Y}_\mathbf{j}^q$ are two sequence of facial motions. Here, $\mathbf{i}, \mathbf{j}$ are the frame indices in sequences, and p, q indicate subject-specific styles.

the decomposition network and audio-to-motion network respectively. As the two types of data are paired, the facial motions pronounce the same speech content conveyed from the audio. Naturally, the speech content latent codes $\mathbf{C_i}$ (from the content encoder E_C) and audio latent codes $\mathbf{A_i}$ (from the audio encoder E_A) should have the same semantic meaning. Since the decoder is shared between two networks and the style code s is identical for the paired data (encoded by the style encoder E_S), when we use the same ground truth to supervise outputs from two networks, we give an indirect but strong constraint to E_C and E_A to produce as semantically close latent codes as possible. In this way, we let the decomposition network be further aware of the distinction between style and content information, because the audio latent codes $\mathbf{A_i}$ contain speech content information only but nothing about facial motion style. The loss function for this mechanism will be explained in Section 4.5.5.

4.5.3 Loss Function for Sequence Prediction

We first introduce the loss function for sequence prediction, which is the base function used to compute other loss terms.

For each predicted frame, the loss function consists of a position term, a motion term, and a lip height term. The position term L_t^{pos} encourages proper articulation by minimizing the l_2 distance between the prediction and ground truth. The motion term L_t^{mot} uses backward finite differences to encourage temporal smoothness [8], [18]. The lip height term L_t^{lip} helps supervise the relative height between lips, especially the lip closure [35]. These loss terms are defined as:

$$L_t^{pos}(\mathbf{y}_t, \hat{\mathbf{y}}_t) = ||\mathbf{y}_t - \hat{\mathbf{y}}_t||^2, \tag{3}$$

$$L_t^{mot}(\mathbf{y}_t, \hat{\mathbf{y}}_t) = ||(\mathbf{y}_t - \mathbf{y}_{t-1}) - (\hat{\mathbf{y}}_t - \hat{\mathbf{y}}_{t-1})||^2, \tag{4}$$

$$L_t^{lip}(\mathbf{y}_t, \hat{\mathbf{y}}_t) = ||\mathsf{LipH}(\mathbf{y}_t; \mathbf{I}) - \mathsf{LipH}(\hat{\mathbf{y}}_t; \mathbf{I})||^2, \tag{5}$$

where $\mathsf{LipH}(\mathbf{y}_t; \mathbf{I})$ approximates the average lip relative height by computing the y-axis difference of pre-selected vertices in the facial mesh $\mathbf{v}_t = \mathbf{y}_t + \mathbf{I}$. $\mathbf{I}$ is the identity shape geometry as mentioned in Section 4.2. The loss function at t-th frame is defined as:

$$L_t(\mathbf{y}_t, \hat{\mathbf{y}}_t) = L_t^{pos}(\mathbf{y}_t, \hat{\mathbf{y}}_t) + \lambda_m \cdot L_t^{mot}(\mathbf{y}_t, \hat{\mathbf{y}}_t) + \lambda_l \cdot L_t^{lip}(\mathbf{y}_t, \hat{\mathbf{y}}_t), \tag{6}$$

where λ_m and λ_l are scaling weights. Finally, the sequential loss is averaged over all frames in a sequence:

$$L_{seq}(\mathbf{Y_i}, \widehat{\mathbf{Y}}_\mathbf{i}) = \frac{1}{|\mathbf{i}|}\sum_{t\in\mathbf{i}} L_t(\mathbf{y}_t, \hat{\mathbf{y}}_t). \tag{7}$$

4.5.4 Loss Function for Decomposition Network

Before formulating the entire joint training loss function, we introduce the loss terms for decomposition network first.

The basic loss function of the encoder-decoder like decomposition network is to recover the inputs by decoding the encoded style and content, which is defined as:

$$L_{rec} = L_{seq}\big(\mathbf{Y}_\mathbf{i}^p, D\big(E_S(\mathbf{Y}_\mathbf{i}^p), E_C(\mathbf{Y}_\mathbf{i}^p)\big)\big). \tag{8}$$

However, we cannot train the network naively as an autoencoder, which is explained before. Thus, we apply the novel two-pass style swapping. More formally, let us denote two training data as $\mathbf{Y}_\mathbf{i}^p$ and $\mathbf{Y}_\mathbf{j}^q$ where $\mathbf{i}, \mathbf{j}$ are the frame indices in sequences, and p, q indicate corresponding styles. For the first-pass style swapping, we have:

$$\mathbf{s}_p = E_S(\mathbf{Y}_\mathbf{i}^p), \quad \mathbf{C_i} = E_C(\mathbf{Y}_\mathbf{i}^p), \tag{9}$$
$$\mathbf{s}_q = E_S(\mathbf{Y}_\mathbf{j}^q), \quad \mathbf{C_j} = E_C(\mathbf{Y}_\mathbf{j}^q), \tag{10}$$
$$\widehat{\mathbf{Y}}_\mathbf{i}^q = D(\mathbf{s}_q, \mathbf{C_i}), \ \widehat{\mathbf{Y}}_\mathbf{j}^p = D(\mathbf{s}_p, \mathbf{C_j}). \tag{11}$$

For the second-pass style swapping, we have:

$$\mathbf{s}'_q = E_S(\widehat{\mathbf{Y}}_\mathbf{i}^q), \quad \mathbf{C}'_\mathbf{i} = E_C(\widehat{\mathbf{Y}}_\mathbf{i}^q), \tag{12}$$
$$\mathbf{s}'_p = E_S(\widehat{\mathbf{Y}}_\mathbf{j}^p), \quad \mathbf{C}'_\mathbf{j} = E_C(\widehat{\mathbf{Y}}_\mathbf{j}^p), \tag{13}$$
$$\widehat{\mathbf{Y}}_\mathbf{i}^{\prime p} = D(\mathbf{s}'_p, \mathbf{C}'_\mathbf{i}), \ \widehat{\mathbf{Y}}_\mathbf{j}^{\prime q} = D(\mathbf{s}'_q, \mathbf{C}'_\mathbf{j}). \tag{14}$$

As for supervision over outputs from the first-pass style swapping, we have to discuss two different cases. First, when $p = q$, i.e., the two data have the same style, we can simply use the inputs as ground truth. Second, when $p \neq q$, i.e., the two data have different styles, we do not have easy access to the ground truth. Note we are using VOCASET as the auxiliary data, where some sentences are shared among different subjects. Hence, for those sentences spoken by different subjects, we can align them with the Dynamic Time Warping (DTW) algorithm [41] over the spectrograms of audio signals. Since the identity information is removed by subtracting identity shape geometry [4], [5], [6], [8], [15], the facial motions (vertex displacements) in aligned sequences have identical speech content but different subject styles regardless of identity, which can be used in the case $p \neq q$. Considering both cases, the loss term is defined as:

$$L_{swp} = \begin{cases} L_{seq}(\mathbf{Y}_\mathbf{i}^p, \widehat{\mathbf{Y}}_\mathbf{i}^q) + L_{seq}(\mathbf{Y}_\mathbf{j}^q, \widehat{\mathbf{Y}}_\mathbf{j}^p), & \text{if } p = q \\ L_{seq}(\mathbf{Y}_\mathbf{i}^{q\to p}, \widehat{\mathbf{Y}}_\mathbf{i}^q) + L_{seq}(\mathbf{Y}_\mathbf{j}^{p\to q}, \widehat{\mathbf{Y}}_\mathbf{j}^p), & \text{otherwise.} \end{cases} \tag{15}$$

where $\mathbf{Y}_\mathbf{i}^{q\to p}$ means the data of style q aligned to data of style p for frame indices $\mathbf{i}$ by DTW. However, the aligned data is not available for all training data, especially for the target subject. We control the sampling method for training mini-batches to make sure L_{swp} can always be computed as described in Section 5.1.1. It is worth mentioning that even though the DTW-aligned data only exist for the subjects in auxiliary data, this loss term still helps the networks properly disentangle the style and content for the target subject's data as shown by an ablation study in Section 5.5.2.

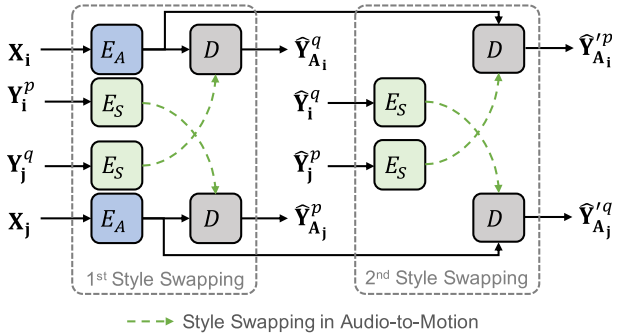


Fig. 7. The two-pass style swapping in the audio-to-motion network during joint training. $\mathbf{X_i}$ is the audio feature paired with $\mathbf{Y}_\mathbf{i}^p$. $\mathbf{X_j}$ is the audio feature paired with $\mathbf{Y}_\mathbf{j}^q$. The content encoder E_C is omitted to give a clear visualization.

For the outputs of the second-pass style swapping, we can use cycle-consistency as supervision, as the ground truth is just the inputs. It is defined as:

$$L_{cyc} = L_{seq}(\mathbf{Y}_\mathbf{i}^p, \widehat{\mathbf{Y}}_\mathbf{i}^{\prime p}) + L_{seq}(\mathbf{Y}_\mathbf{j}^q, \widehat{\mathbf{Y}}_\mathbf{j}^{\prime q}). \tag{16}$$

Finally, the overall loss function for decomposition network is:

$$L_{decomp} = \lambda_{rec} \cdot L_{rec} + \lambda_{swp} \cdot L_{swp} + \lambda_{cyc} \cdot L_{cyc}. \tag{17}$$

Here λ_{rec}, λ_{swp}, and λ_{cyc} are scaling weights.

4.5.5 Loss Function for Joint Training

When we jointly train the audio-to-motion and decomposition networks, paired data are fed into the two networks. The two-pass style swapping is also applied in the audio-to-motion network as shown in Fig. 7. $\mathbf{X_i}$, paired with $\mathbf{Y}_\mathbf{i}^p$, is encoded by the audio encoder E_A to get audio latent codes. The first-pass style swapping is done by injecting the style latent code encoded from $\mathbf{Y}_\mathbf{j}^q$, drawn as a green dotted arrow. The second pass is performed similarly. To be formal:

$$\mathbf{A_i} = E_A(\mathbf{X_i}), \quad \mathbf{A_j} = E_A(\mathbf{X_j}), \tag{18}$$
$$\widehat{\mathbf{Y}}^q_{\mathbf{A_i}} = D(\mathbf{s}_q, \mathbf{A_i}), \quad \widehat{\mathbf{Y}}^p_{\mathbf{A_j}} = D(\mathbf{s}_p, \mathbf{A_j}), \tag{19}$$
$$\widehat{\mathbf{Y}}^{\prime p}_{\mathbf{A_i}} = D(\mathbf{s}'_p, \mathbf{A_i}), \quad \widehat{\mathbf{Y}}^{\prime q}_{\mathbf{A_j}} = D(\mathbf{s}'_q, \mathbf{A_j}). \tag{20}$$

As described before in Section 4.5.2, we want to use the same ground truth to supervise both networks to ensure the content encoder E_C produces semantically close latent codes to those from the audio encoder E_A, so as to further emphasize the difference between style and content information. Hence, we compute the loss function for the audio-to-motion network similarly to the decomposition network by replacing $E_C(\mathbf{Y})$ with the paired $E_A(\mathbf{X})$ in equation 8, replacing $\widehat{\mathbf{Y}}_\mathbf{i}^q, \widehat{\mathbf{Y}}_\mathbf{j}^p$ with $\widehat{\mathbf{Y}}^q_{\mathbf{A_i}}, \widehat{\mathbf{Y}}^p_{\mathbf{A_j}}$ in equation 15, and replacing $\widehat{\mathbf{Y}}_\mathbf{i}^{\prime p}, \widehat{\mathbf{Y}}_\mathbf{j}^{\prime q}$ with $\widehat{\mathbf{Y}}^{\prime p}_{\mathbf{A_i}}, \widehat{\mathbf{Y}}^{\prime q}_{\mathbf{A_j}}$ in equation 16 to get equivalent loss

TABLE 1
Collected Videos

Subject	Training	Validation	Source
Donald Trump[1]	109 s	53 s	Internet
Barack Obama[1]	89 s	61 s	Internet
Hillary Clinton[1]	76 s	55 s	Internet
Prosper Taruvinga[2]	54 s	30 s	Internet
Actor A[3]	54 s	34 s	iPhone X

[1] *The videos are in Public Domain.*
[2] *The video entitled "Free Website Audit" is owned by Prosper Taruvinga from* Livelong Digital *(CC-BY 3.0).*
[3] *The video is captured by us with an iPhone X front-facing camera.*

terms L^A_{rec}, L^A_{swp}, and L^A_{cyc}, which are also weighted summed as:

$$L_{audio} = \lambda^A_{rec} \cdot L^A_{rec} + \lambda^A_{swp} \cdot L^A_{swp} + \lambda^A_{cyc} \cdot L^A_{cyc}. \quad (21)$$

Here λ^A_{rec}, λ^A_{swp}, and λ^A_{cyc} are scaling weights.

We have also tried to apply a direct constraint on the latent codes **C** and **A** by minimizing l_2 distance between them. Nevertheless, the model with such extra constraint only shows imperceptible improvement. It indicates that our strategies, i.e., two-pass style swapping and joint training with decoder sharing, are sufficient for our task. For simplicity, we do not add the extra loss over latent codes.

The final loss function for jointly training the networks hence is:

$$L_{joint} = L_{decomp} + L_{audio}. \quad (22)$$

4.6 Generation

After training, we send one entire training sequence of reconstructed facial motions from the target subject into the decomposition network to get a disentangled style code. Afterward, for any audio inputs, we can use the audio-to-motion network to generate personalized 3D facial animations with the disentangled style code.

5 Experimental Results

We have conducted extensive experiments to investigate the ability of our proposed approach on video data from five subjects, separately. Experimental results have shown that our model outperforms previous state-of-the-art methods qualitatively and quantitatively.

5.1 Implementation Details

For each subject, we collect a short video clip lasting about one to two minutes, at 25 frames per second (FPS), from the Internet or captured by a smartphone camera. And we train models respectively using the 3D reconstruction of video data together with the auxiliary data from VOCASET. The data of VOCASET is originally captured at 60 FPS. To be matched with the video data, sequences from VOCASET are linearly downsampled to 25 FPS. While all data of VOCASET are used during training, the reconstructed data from subjects are split into disjoint training and validation sets as shown in Table 1.

TABLE 2
Hyper-Parameters for the Audio Encoder

Layer	d_{ff}	n_{head}	d_{model}	Activation	Dropout
Transformer Encoding	2048	4	64	ReLU	0.1
Transformer Encoding	2048	4	64	ReLU	0.1
Transformer Encoding	2048	4	64	ReLU	0.1
Transformer Decoding	2048	4	64	ReLU	0.1

TABLE 3
Hyper-Parameters for the Causal Sequential Module

Layer	Kernel Size	Feature Maps	Activation	Dropout
Conv 1D	5	64	LReLU 0.2[a]	0.1
Conv 1D	3	128	LReLU 0.2	0.1
Conv 1D	3	256	LReLU 0.2	0.1

[a] *LReLU 0.2: LeakyReLU with 0.2 slope rate for negative part.*

5.1.1 Batch Sampling

Our model is trained by the standard Adam optimization algorithm [42] with mini-batches. Since all training sequences, including the auxiliary ones, have different lengths, varying from several seconds to one minute, it is not wise to fulfill a mini-batch with entire sequences padded to the same length. Instead, we use *sub-sequences* of 20 frames in length. Each sequence is split into overlaid sub-sequences by hopping 5 frames each time. While being trained on sub-sequences, the decomposition network can disentangle a style latent code properly when we feed an entire training sequence as input during generating phase.

Another issue we have to consider is that supervision data are required for loss terms L_{swp}, L^A_{swp} as described before. Since only a part of VOCASET data has DTW-aligned peers, we have to sample mini-batches carefully. For a batch in size $B = 4$, we sample $\frac{B}{2} = 2$ sub-sequences randomly at first. Then, for each sampled sub-sequence, if the DTW-aligned peers exist, we randomly sample one out of the peers ($p \neq q$); otherwise, we randomly sample another sub-sequence from the same sequence ($p = q$).

5.1.2 Hyper-Parameters

We use the same hyper-parameters in all models. For audio features, we use $W = 16$ frames of DeepSpeech features in a temporal window, giving 320ms short-term audio context. For the audio-to-motion network, the hyper-parameters are listed in Tables 2, 3, and 4. In the causal sequential module, the stacked convolution layers have a receptive field of 9 frames, i.e., 8 history frames are taken into consideration. The hyper-parameters of the style and content encoders in the decomposition network are listed in Table 5. Both

TABLE 4
Hyper-Parameters for the Displacement Decoding Module

Layer	Feature Maps	Activation
FC[a]	256	LReLU 0.2
FC	256	LReLU 0.2
FC	128	LReLU 0.2
FC	50	-
FC	5023×3	-

[a] *FC: Fully connected layer.*

TABLE 5
Hyper-Parameters for the Style/Content Encoder

Layer		Length	Feature Maps	Activation	Downsample
Spiral Conv		12	16	LReLU 0.2	1/4
Spiral Conv		12	32	LReLU 0.2	1/4
Spiral Conv		12	32	LReLU 0.2	1/4
FC	(in E_C)	-	64	LReLU 0.2	-
	(in E_S)	-	32	LReLU 0.2	-
LSTM	(in E_S)	-	32	-	-

encoders use a three-layers SpiralNet [40] with the same structure but do not share the weights. The decoder is reused from the audio-to-motion network.

For the loss function of a single predicted frame in equation 6, we use $\lambda_m = 5$ and $\lambda_l = 1$. For the weights of loss terms to supervise the decomposition network in equation 17, we use $\lambda_{rec} = 1$, $\lambda_{swp} = 3$, and $\lambda_{cyc} = 1$. For the audio related loss terms in equation 21, we use $\lambda^A_{rec} = 2$, $\lambda^A_{swp} = 6$, and $\lambda^A_{cyc} = 2$. We train each model for 50 epochs by using an Adam [42] optimizer with 0.0001 learning rate. An exponential method is applied to decay the learning rate by 0.9 every epoch starting after 30 epochs.

5.2 Photo-Realistic Video Synthesis

A simple application for personalized audio-driven 3D facial animations is the synthesis of photo-realistic talking face videos, which is shown by previous methods [4], [5], [6]. To confirm that our proposed approach helps improve the photo-realistic video synthesis, we train a deferred neural rendering network [43] for each subject to synthesize videos from generated 3D facial animations. In a deferred neural rendering network, a neural texture for the corresponding subject is sampled by UV coordinates rasterized from a facial 3D mesh into the image plane. The sampled neural texture is rendered by a Pix2PixHD generator [44] into a photo-realistic face image. Another Pix2PixHD generator follows to blend the rendered face image into a background. The networks are trained with the l_1 and VGG-based perceptive loss terms. For the details, please see [4], [6], [43].

5.3 Qualitative Comparison

We compare our approach with four previous works [4], [5], [6], [8]. Considering some of them [4], [5], [6] use different video datasets and reconstruct 3D facial meshes with the Basel Face Model (BFM), which has a different topology from the FLAME we used, we compare ours with these works in two ways.

Our data. First, to give a fair comparison, we train their models [4], [5], [6], [8] from scratch on the same data as ours, i.e., our reconstructed data from a target subject and the auxiliary data from VOCASET, using their officially released codes. However, VOCA [8] requires data at 60 FPS, different from 25 FPS preferred by ours and the other three methods. We upsample reconstructed data to 60 FPS to train VOCA and downsample the generated results to 25 FPS again to give a fair comparison. In this case, we also use the same deferred neural rendering networks trained by us to render photo-realistic videos for 3D results from all methods. Fig. 8 shows some key frames of results, where we input audio from validation data, which are unseen during training. Our results are closest to the ground truth at the pronunciation of bilabial plosive phonemes /p/, where lips are supposed to be pressed. Whereas, some other methods may completely fail to close lips at such critical phonemes.

Pre-trained. Second, we also use their [4], [5], [6] BFM trackers to reconstruct 3D data and learn a new style based on their official pre-trained models by inferring a mapping matrix [4], fine-tuning the pre-trained model [5], or handcrafting a style code from the statistics of reconstructed data [6]. Since our approach uses a different mesh topology from theirs, we do not compare mesh results in this case. Instead, we only compare our photo-realistic results with theirs rendered by their own rendering networks. As for VOCA [8], it is originally just trained on VOCASET, so it is not considered in this case. Some key frames of the results on validation data are shown in Fig. 9. It turns out that even inferred or fine-tuned on their pre-trained models, other methods may still fail to pronounce some bilabial phonemes /m/, /b/, and /p/, which our approach succeeds to utter. It is worth mentioning that the scales of their original auxiliary data are much larger than ours (VOCASET).

In short, we notice that these previous efforts may fail to pronounce some phonemes articulately especially **/b/, /p/, and /m/**, even though they can learn the facial motion style from a specific subject. Our approach, however, can generate results not only under the target subject's motion style but also more robust than others in such difficult cases.

We also give some results of different subjects when the same test audio clip is inputted as in Fig. 10. For more results and dynamic comparison, please watch our supplementary video.

5.4 Quantitative Comparison

We also compare with previous works on the validation data quantitatively.

In the first case where models are trained on our data, we use the 3D reconstruction data as ground truth to measure errors of mesh predictions. Besides, the photo-realistic videos rendered from 3D results of different methods by the same (our) deferred neural rendering networks are compared with real videos.

In the second case where their official pre-trained models and BFM trackers are used, we do not compute errors on meshes because each method has different 3D reconstruction results. We only compare the synthesized videos, which are rendered by the respective rendering network of each method. As the pre-trained neural rendering network of Wu et al. [6] does not support background generation, we use the background from the real videos directly, which should let their method gain a little advantage when measuring metrics.

For the mesh results, we consider the correctness of not only the facial motions but also the lip motions, because motions of lips have a critical impact on the presentation of style and the naturalness of speech synchronized animation. For each frame, we measure the Average Vertex Distance of lip area (AVD-L) and face area (AVD-F) between generated

Fig. 8. Comparison with previous works trained on our data. Some frames of results generated from the validation set are shown. The output meshes of each subject are rendered to photo-realistic images by the same neural rendering network. The first column is the ground truth, i.e., real video frames and meshes reconstructed from videos. The other columns show results of all methods trained on our data. Each predicted mesh is accompanied by an error heatmap. Green rectangles highlight visually correct lip motions and red ones highlight wrong lip motions. Donald Trump is saying /p/ in "process". Barack Obama is saying /p/ in "opportunity". Hillary Clinton is saying /p/ in "keep". Prosper Taruvinga is saying /p/ in "page". Actor A is saying /p/ in "strip".

Fig. 9. Comparison with previous works using their official pre-trained models and their BFM reconstruction data. Some frames of results generated from validation set are shown. The images are rendered by their own rendering networks. The comparison of meshes is omitted due to different topologies. Donald Trump is saying /p/ in "support". Barack Obama is saying /b/ in "job". Hillary Clinton is saying /m/ in "America". Prosper Taruvinga is saying /m/ in "shame". Actor A is saying /p/ in "strip".

and reconstructed meshes. Eyeballs and eyelids are ignored when calculating AVD-F since eye movements are almost irrelevant to speech. We average AVD-L and AVD-F across all validation frames and the numerical results are listed in Table 6. As a smaller vertex distance means better prediction, our model outperforms other methods, especially in the lip area.

For the video results, we measure the traditional Peak Signal-to-Noise Ratio (PSNR) values between generated and real videos to judge the per-frame quality. Higher PSNR means better per-frame quality. In addition, we use the Fréchet Video Distance (FVD) [45] to measure the overall perceptual distances between the generated and real videos. Lower FVD means better visual quality for humans. The numerical results are listed in Table 7. Our approach gets the best FVD and the second-best PSNR. Models of Yi et al. [5] (pre-trained) get the best PSNR values because their GAN-based rendering network is pre-trained on large-scale video datasets, while other methods, including ours, use deferred neural rendering networks trained on fewer data. Even though Yi et al. [5] (pre-trained) can generate images with the best per-frame quality, their generated videos suffer from temporal instability, which makes the videos perceptually unrealistic and leads to worse FVD values than ours.

5.5 Ablation Study

5.5.1 Ablation Study of Style Source

The key contribution of our approach is that we use a decomposition network to disentangle style and content information

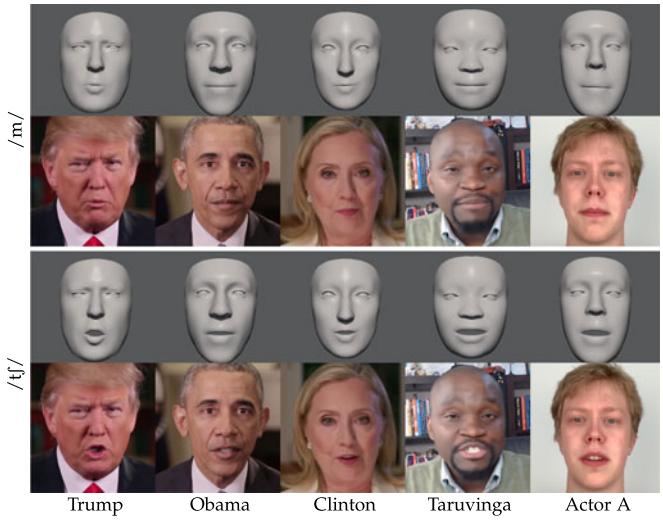


Fig. 10. Visualization of our generated animations from different subjects when the same test audio clip is given. The subjects are saying /m/ and /tʃ/ in the word "much".

from facial motions and use the disentangled style latent code to guide the generation of personalized animations. We study the necessity of the decomposition network as a source of the style code by testing three alternatives: 1) a baseline model, which only contains an audio-to-motion network without any style code, 2) replacing the decomposition network with a naive style encoder to encode style codes from facial motions, and 3) a style embedding layer used as a replacement for the decomposition network to provide style codes. The baseline model is trained on reconstructed 3D data only, due to the lack of ability to distinguish styles. Either the naive style encoder or the style embedding layer is trained jointly with an audio-to-motion network in an end-to-end way on the reconstructed and auxiliary data, without the aforementioned two-pass style swapping mechanism necessary for the decomposition network. The naive style encoder shares the same structure as the style encoder E_S in our decomposition network. The style embedding layer is simply a look-up table for style codes according to subject labels. Both of them produce style codes in the same dimension as our decomposition network.

Our approach generates 3D animations with better pronouncing accuracy on the validation set than the baseline model as shown in Table 8. It proves that our approach can properly utilize the auxiliary data to improve speech robustness. The naive style encoder alternative may perform worse than the baseline and fail to learn proper styles, especially for Donald Trump, Barack Obama, and Prosper Taruvinga, as shown in Table 8 and Fig. 11. It means that this alternative cannot properly distinguish styles between different subjects, and the auxiliary data become a burden for learning the target subject's motion style. The possible reason is that it cannot differentiate motion style from speech content in the input facial motions and treat some content information as style.

The style embedding alternative has close metrics with our full model on the validation set in Table 8. Whereas, it has some failures in modeling the target subject's motion style when we analyze it qualitatively. A simple but effective way to test if a model solidly learns the target subject's style

TABLE 6
Mesh Quantitative Comparison on Validation Set With Previous Methods Trained on Our Data

	Donald Trump		Barack Obama		Hillary Clinton		Prosper Taruvinga		Actor A	
	AVD-L ↓[a]	AVD-F ↓[b]	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓
VOCA [8] (our data)	2.611	1.051	1.984	0.978	2.301	1.074	3.275	1.272	1.799	0.825
NVP [4] (our data)	2.869	1.113	2.262	1.062	2.431	1.098	3.244	1.278	1.859	0.843
Yi et al.[5] (our data)	3.257	1.251	2.466	1.136	2.777	1.231	4.114	1.573	2.215	0.963
Wu et al.[6] (our data)	3.154	1.213	2.621	1.196	2.616	1.188	4.059	1.525	2.160	0.953
Ours	**2.421**	**0.978**	**1.898**	**0.954**	**2.212**	**1.044**	**3.138**	**1.252**	**1.699**	**0.825**

[a] *AVD-L: Average Vertex Distance of Lip area. Unit is millimeter. Lower is better.*
[b] *AVD-F: Average Vertex Distance of Face area. Unit is millimeter. Lower is better.*

TABLE 7
Video Quantitative Comparison on Validation Set With Previous Methods

	Donald Trump		Barack Obama		Hillary Clinton		Prosper Taruvinga		Actor A	
	PSNR ↑[a]	FVD ↓[b]	PSNR ↑	FVD ↓	PSNR ↑	FVD ↓	PSNR ↑	FVD ↓	PSNR ↑	FVD ↓
VOCA [8] (our data)	31.2	208	31.9	117	35.0	122	30.2	159	35.2	228
NVP [4] (our data)	30.9	219	31.8	128	34.8	118	30.1	171	35.0	223
Yi et al.[5] (our data)	30.5	208	31.7	137	34.5	136	29.4	166	34.3	231
Wu et al.[6] (our data)	30.5	218	31.5	170	34.5	149	29.5	181	34.6	226
NVP [4] (pre-trained)	26.7	268	29.4	179	30.7	183	27.0	173	30.5	355
Yi et al.[5] (pre-trained)	**33.3**	221	**33.3**	178	**35.3**	140	**30.6**	190	34.3	228
Wu et al.[6] (pre-trained)	30.4	300	31.1	156	31.9	139	27.1	182	31.1	330
Ours	31.3	**173**	32.0	**106**	35.0	**114**	30.3	**147**	**35.3**	**214**

[a] *PSNR: Peak Signal-to-Noise Ratio. Unit is dB. Higher is better.*
[b] *FVD: Fréchet Video Distance. Lower is better.*

TABLE 8
Mesh Quantitative Comparison on Validation Set With Ablations of Style Source

	Donald Trump		Barack Obama		Hillary Clinton		Prosper Taruvinga		Actor A	
	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓
Baseline (No Style Code)	2.620	1.042	2.154	0.997	2.338	1.070	3.842	2.283	1.851	0.831
Naive Style Encoder	4.586	1.586	5.720	2.556	2.368	1.152	4.801	2.125	1.974	0.849
Style Embedding	2.467	1.001	1.978	0.989	2.221	**1.041**	3.525	2.236	**1.656**	**0.774**
Full (Decomposition)	**2.421**	**0.978**	**1.898**	**0.954**	**2.212**	1.044	**3.138**	**1.252**	1.699	0.825

is to input audio from an auxiliary subject and generate animations under the target subject's style. We have to observe whether the motion style of generated results is affected by the auxiliary subject. As shown in Fig. 12, when we feed an audio clip from an auxiliary training subject to the style embedding alternative, its results may mix styles of both the target subject and the auxiliary subject in some cases. Because we hope to learn the motion style of the target subject rather than those of auxiliary training data, the mixture of styles is treated as a failure. For dynamic results, please watch our supplementary video.

5.5.2 Ablation Study of Loss Terms

Since our final loss function used to train the entire framework is composed of many terms, we investigate their necessities as well.

To confirm the indispensability of terms L_{swp}, L^A_{swp} and the strategy to compute them with DTW-aligned data, we conduct two experiments. In the first, we simply remove L_{swp}, L^A_{swp} from the loss function. In the second, we only feed sub-sequences with the same style to compute L_{swp}, L^A_{swp}, i.e., without the case of $p \neq q$ in them. As shown in Table 9, without L_{swp}, L^A_{swp}, models get inferior metrics. Without the case of $p \neq q$, models have close metrics, sometimes better. However, both ablations have a similar style mixing issue with the style embedding alternative as described before and shown in Fig. 12.

We remove the loss terms L_{cyc}, L^A_{cyc}, to see if the cycle-consistency constraint is useful. It turns out that the absence leads to slightly inferior metrics in lip area for four subjects (Table 9) and less robust results at bilabial phonemes in foreign languages (Fig. 13).

When we naively jointly train two networks end-to-end without two-pass style swapping, i.e., only with L_{rec}, L^A_{rec}, the models may fail to learn proper target motion styles as shown in Fig. 14. Besides, significant deterioration of metrics can be observed on the validation set for some subjects (Table 9). For dynamic results, please watch our supplementary video.

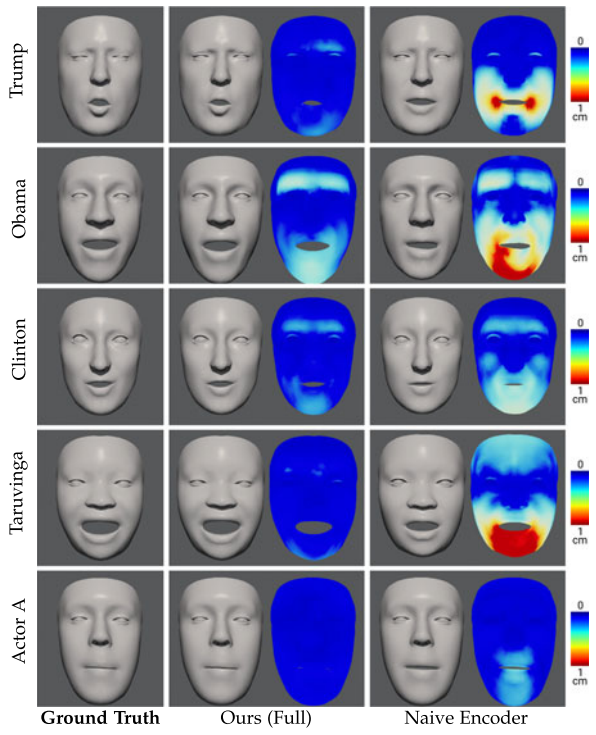


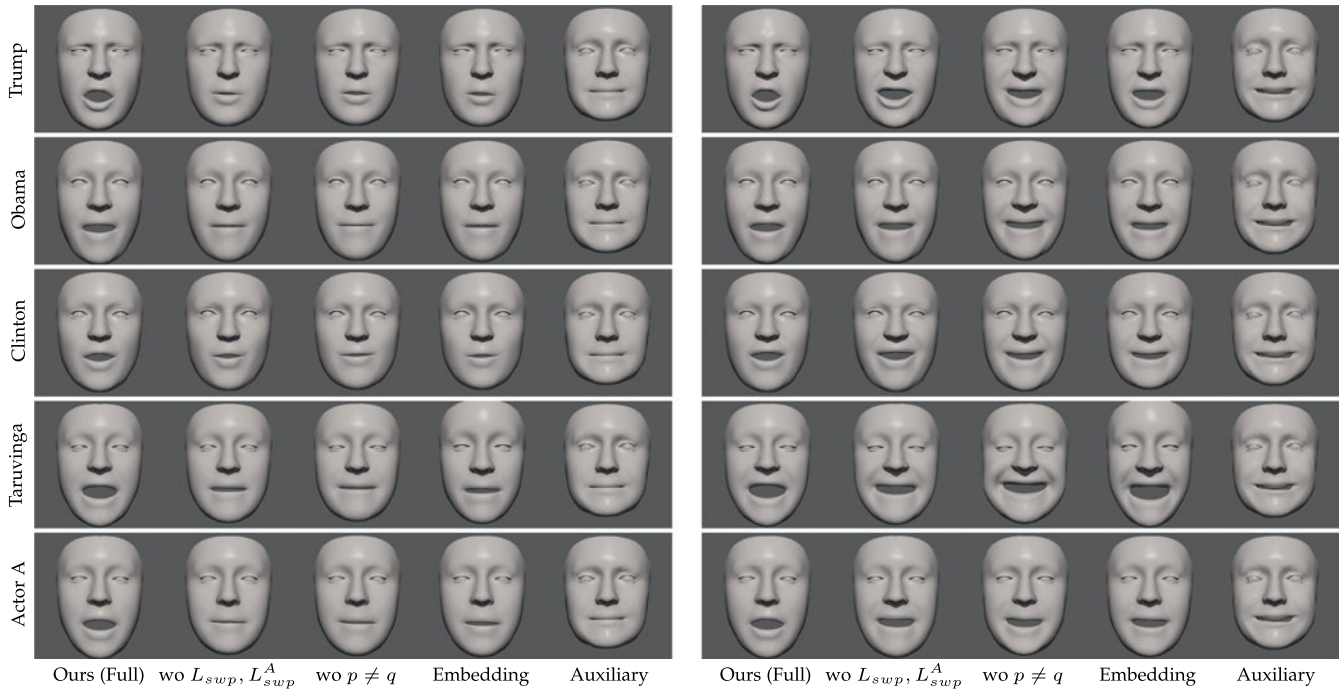
Fig. 11. Ablation study of style source, in which a naive style encoder is used to replace the decomposition network. Some frames from the validation set for each subject are shown. Obvious mismatch of mouth shape can be observed in the results from the naive style encoder (the last column). The heatmaps show the errors compared with the ground truth.

5.6 Visualization of Disentangled Information

It is interesting to visualize the disentangled style and content information. We input a validation sequence from Donald Trump into his trained decomposition network to get disentangled style and content latent codes. We can entangle the latent codes again to reconstruct the input sequence. Besides, we can generate a "content-only" sequence by decoding the disentangled content latent codes with an all-zero style code. Similarly, a "style-only" sequence can be decoded from the disentangled style latent code with all-zero content codes. Some key frames are shown in Fig. 15. We make an observation that the "content-only" animation delivers speech content without the subject's style, and the "style-only" one keeps almost unchanged to depict the subject-specific style in a non-speaking state. Besides, if we input two motion sequences with different contents from the same subject, we can observe that the "style-only" animations are still almost the same (Fig. 16), which indicates a consistent disentanglement of style for the same person with our method. However, the "style-only" animations will be quite different if the input motion sequences are from different subjects, even if they are pronouncing the same phonemes (Fig. 17). Please see our supplementary video for dynamic results.

5.7 Impact of Different Input Modalities on Motion Style

During generation, our approach requires two kinds of inputs: an audio clip used to drive the talking animation and a motion sequence used to disentangle style code for



(a) Frame 1: Preparing to pronounce '/h/' in word 'headgear'.

(b) Frame 2: Pronouncing '/g/' in word 'headgear'.

Fig. 12. Two frames are shown for a case where an audio clip from auxiliary training data is fed to generate animations uttering word 'headgear'. To demonstrate styles better, all facial motions are added to the same mean "zero-pose" template $\overline{\mathbf{T}}$. In both frames, results from ablation without L_{swp}, L^A_{swp} (column 2), ablation without $p \neq q$ in L_{swp}, L^A_{swp} (column 3), and ablation using style embedding layer (column 4) are affected by the style of auxiliary subject (column 5). In (a) frame 1, ablation models follow the auxiliary subject's style and keep mouth almost closed. In (b) frame 2, ablation models follow the auxiliary subject's style and grin exaggeratedly. While models of different subjects are trained separately, similar failures appear in all cases.

TABLE 9
Mesh Quantitative Comparison on Validation Set With Ablations of Loss Terms

	Donald Trump		Barack Obama		Hillary Clinton		Prosper Taruvinga		Actor A	
	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓	AVD-L ↓	AVD-F ↓
wo Two-Pass Swapping	5.074	2.030	3.229	1.426	4.630	1.841	3.226	1.288	2.495	1.059
wo L_{cyc}, L^A_{cyc}	2.449	**0.973**	1.935	0.968	2.235	1.041	**3.069**	**1.219**	1.706	0.806
wo L_{swp}, L^A_{swp}	2.623	1.077	1.928	0.991	2.414	1.164	3.203	1.253	1.712	0.836
wo $p \neq q$ in L_{swp}, L^A_{swp}	2.432	0.975	1.987	0.986	**2.186**	**1.024**	3.150	1.232	1.710	**0.798**
Full (All Terms)	**2.421**	0.978	**1.898**	**0.954**	2.212	1.044	3.138	1.252	**1.699**	0.825

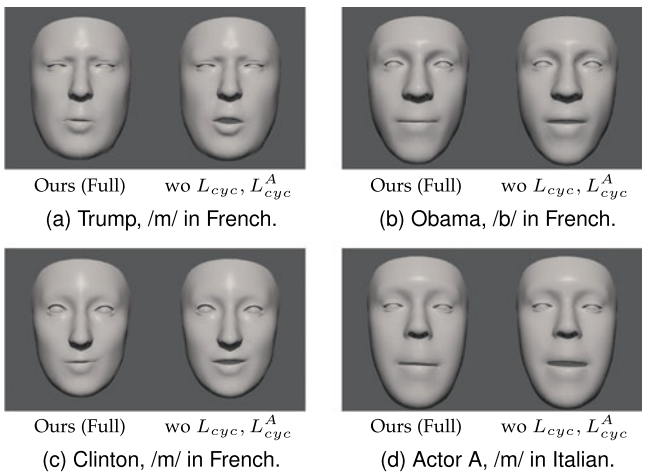


Fig. 13. Ablation study of loss terms, in which loss terms L_{cyc}, L^A_{cyc} are removed. As shown by some frames of bilabial phonemes from foreign language test audio clips, models may fail to press lips in some cases without cycle-consistency constraint.

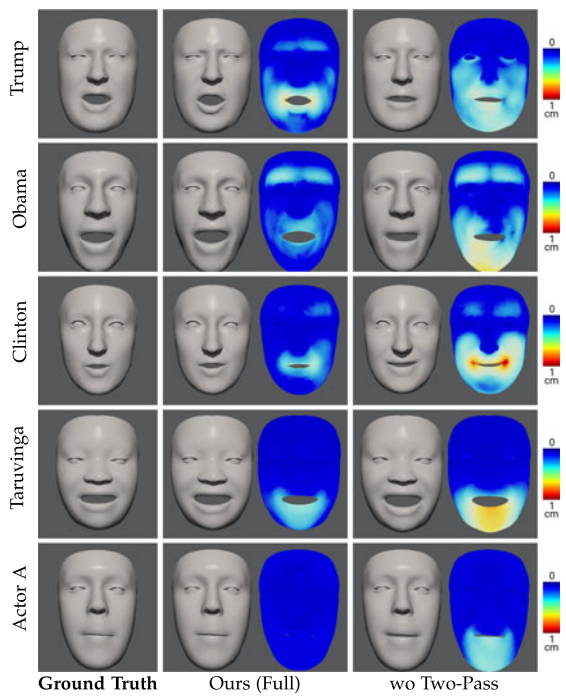


Fig. 14. Ablation study of loss terms, in which two-pass style swapping is not used. As shown by some frames from the validation set for each subject, obvious mismatch of mouth shapes can be observed if trained without two-pass style swapping. The heatmaps show the errors compared with the ground truth.

guiding the personalized animation generation. We attempt to study how they affect the motion style, respectively.

Audio. When we utilize different audio clips saying the same sentence to drive a face, along with the same style code extracted from a motion sequence, the motion style is hardly affected. As shown by Fig. 18a, the "style-only" results are the same and our final outputs are very close. Here, we align the results by the DTW algorithm to compare them since the two audio clips have different time durations.

Motion. To study the effect of input motion sequences, we use the same audio clip input. When we input two motion sequences with different contents from the same subject to provide style codes, we can see that the motion styles of results are almost the same (Fig. 18b). Nevertheless, when the input motion sequences are from different subjects, even if they have the same content, the motion styles of results are quite different (Fig. 18c).

5.8 User Study

We also conduct a user study to compare our approach with previous state-of-the-art methods [4], [5], [6], [8]. We test all

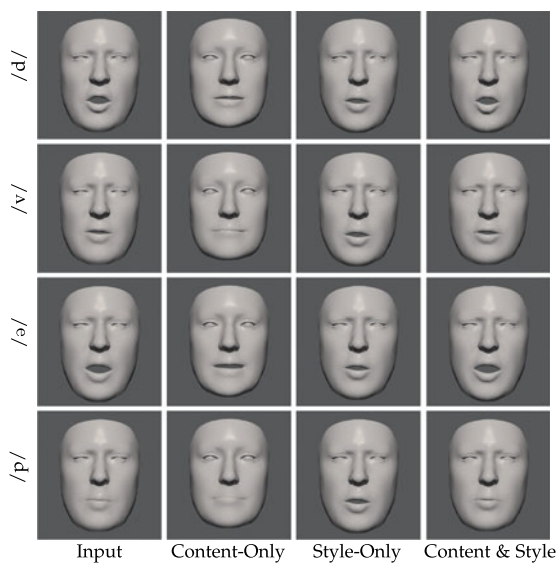


Fig. 15. Visualization of style-content disentanglement. A validation sequence from Donald Trump is disentangled. Key frames during pronouncing the word "develop" are shown row by row. "Content-Only" shows results decoded from the disentangled content codes and an all-zero style code. "Style-Only" shows results decoded from the disentangled style code and all-zero content codes. "Content & Style" shows results decoded from disentangled style and content codes.

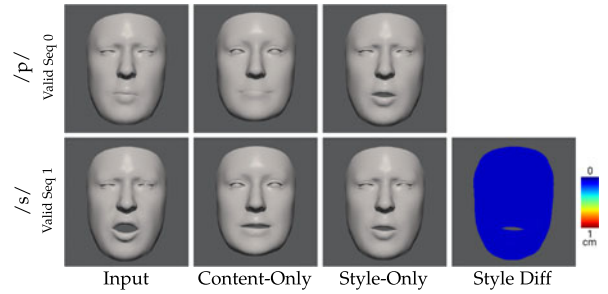


Fig. 16. Visualization of style-content disentanglement of two validation sequences from Donald Trump. In the first row, the subject is saying /p/ in "develop" from one sequence. In the second row, the subject is saying "safety" from the other. The "Style Diff" shows little difference between the "Style-Only" results of two rows, even though the corresponding style codes are disentangled from different sequences.

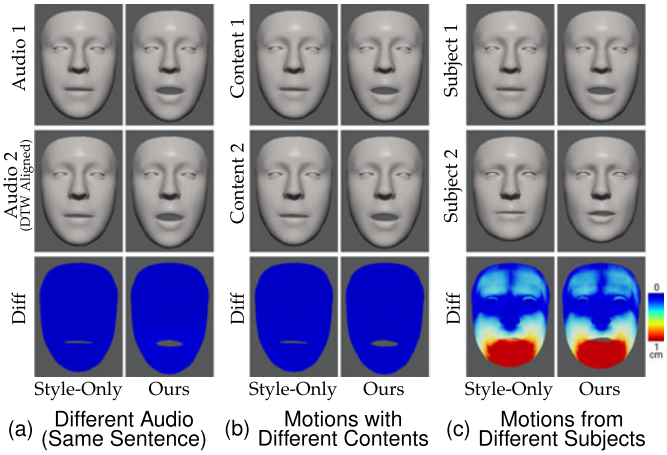


Fig. 18. Visualization of the impact of different input modalities on motion style. In each sub-figure, the first two rows show the results in different input conditions and the last row shows the difference between them. In (a), two different audio clips saying the same sentence and the same motion sequences are inputted. In (b), the same audio clip and two motion sequences with different contents from the same subject are inputted. In (c), the same audio clip and two motion sequences from different subjects saying the same content are inputted. All results are added to the same mean "zero-pose" template $\overline{\mathbf{T}}$. All results are pronouncing /əʊ/ in the word "home".

methods with 15 audio clips (7∼17s) from the Internet, including 5 English speech audio, 5 in noisy environments, and 5 in foreign languages. For each audio clip, generated animation clips from all methods are shown to users side by side as a group. Users evaluate animation clips in three aspects: style similarity with the target subject, facial movement naturalness, and audiovisual synchronization. The score varies from 1 to 5 representing "terrible", "bad", "ok", "good" and "excellent". To measure style similarity, we provide users with a clip of the target subject's talking video from the training set as a reference. In total, we collect opinions from 100 users on Amazon Mechanical Turk. Each user scores 15 animation groups in a randomly selected style. In other words, we collect 1500 opinions in three aspects for each method. The mean opinion scores (MOS) are shown in Table 10. Our approach gets the highest scores in all three aspects.

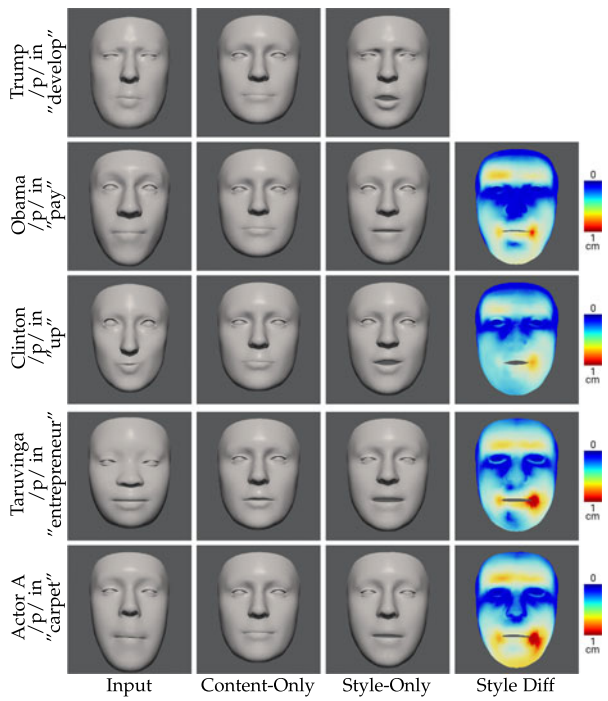


Fig. 17. Visualization of style-content disentanglement of validation sequences from different subjects. To demonstrate content and different styles better, the "Content-Only" and "Style-Only" results are added to the same mean "zero-pose" template $\overline{\mathbf{T}}$. The last column "Style Diff" shows the difference of "Style-Only" results between each row and the first row. We can observe that all subjects press their lips to pronounce /p/ but they have quite different "Style-Only" results.

6 Limitations and Future Work

We briefly discuss the limitations of this work and our future work. One limitation of our approach is that the networks have to be trained from scratch for each new subject. Otherwise, for an out-of-training-distribution subject, our approach may not properly generate stylized animations. Fig. 19 shows this failure case when the data of an unseen subject (from [28]) is given. One future work is to simplify the pipeline of learning the styles from new subjects. Currently, we only focus on the learning of facial motions without head poses in this work. We believe head movements [46], [47], [48] can also be useful in expressing personal styles. We will involve head movements in our future work.

TABLE 10
Mean Opinion Scores of User Study

	Style ↑[a]	Natural ↑[b]	AV-Sync ↑[c]
VOCA [8] (our data)	3.03	2.98	3.01
NVP [4] (our data)	3.08	2.90	2.99
Yi et al.[5] (our data)	2.85	2.66	2.71
Wu et al.[6] (our data)	2.72	2.72	2.60
NVP [4] (pre-trained)	2.89	2.83	2.90
Yi et al.[5] (pre-trained)	2.78	2.70	2.81
Wu et al.[6] (pre-trained)	2.80	2.73	2.77
Ours	**3.22**	**3.09**	**3.23**

[a] *Style similarity with the target subject.*
[b] *Naturalness of facial motions.*
[c] *Audiovisual synchronization.*

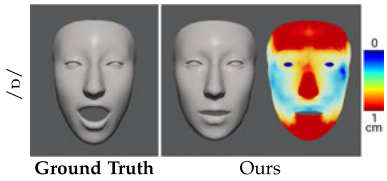


Fig. 19. Visualization of a failure case, in which a motion sequence from an unseen subject is given to disentangle the style code. Our approach fails to generate proper results. The heatmap shows the errors compared with the ground truth.

7 Conclusion

In summary, we present an approach to learning how to generate personalized audio-driven 3D facial animations, robust to unseen speeches, from a target subject's short video. The reconstructed 3D data from the video are used together with the auxiliary VOCASET to improve the model robustness. To achieve more accurate style and articulation, we propose a novel framework, composed of an audio-to-motion network and a novel decomposition network, to disentangle motion style and speech content information from facial motions and naturally use the disentangled style information to distinguish different styles. We propose two novel training mechanisms, including two-pass style swapping and joint training with decoder sharing, to supervise the disentanglement of the style and content information. After training, the audio-to-motion network can be used to generate robust personalized facial animations from unseen speech audio. Extensive experiments have shown that our approach can learn subject-specific motion styles well and have better speech robustness and more appealing visual results compared with previous state-of-the-art methods.

Acknowledgments

The authors would like to thank reviewers for their insightful comments, Cudeiro et al. for publishing VOCASET, and all subjects for sharing videos.

References

[1] T. Karras, T. Aila, S. Laine, A. Herva, and J. Lehtinen, "Audio-driven facial animation by joint end-to-end learning of pose and emotion," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Jul. 2017.

[2] S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, "Synthesizing obama: Learning lip sync from audio," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Jul. 2017.

[3] S. Taylor et al., "A deep learning approach for generalized speech animation," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, Jul. 2017.

[4] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, and M. Nießner, "Neural voice puppetry: Audio-driven facial reenactment," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 716–731.

[5] 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*.

[6] 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.

[7] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero, "Learning a model of facial shape and expression from 4D scans," *ACM Trans. Graph.*, vol. 36, no. 6, pp. 194:1–194:17, 2017.

[8] D. Cudeiro, T. Bolkart, C. Laidlaw, A. Ranjan, and M. J. Black, "Capture, learning, and synthesis of 3D speaking styles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 101–10 111.

[9] A. Hannun et al., "Deep speech: Scaling up end-to-end speech recognition," 2014, *arXiv:1412.5567*.

[10] Z. Deng, M. Bulut, U. Neumann, and S. Narayanan, "Automatic dynamic expression synthesis for speech animation," *Proc. IEEE Comput. Animation Social Agents*, vol. 2004, pp. 267–274, 2004.

[11] Z. Deng, U. Neumann, J. P. Lewis, T.-Y. Kim, M. Bulut, and S. Narayanan, "Expressive facial animation synthesis by learning speech coarticulation and expression spaces," *IEEE Trans. Vis. Comput. Graphics*, vol. 12, no. 6, pp. 1523–1534, Nov./Dec. 2006.

[12] Y. Cao, W. C. Tien, P. Faloutsos, and F. Pighin, "Expressive speech-driven facial animation," *ACM Trans. Graph.*, vol. 24, no. 4, pp. 1283–1302, 2005.

[13] P. Edwards, C. Landreth, E. Fiume, and K. Singh, "JALI: An animator-centric Viseme model for expressive lip synchronization," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–11, Jul. 2016.

[14] L. Song, W. Wu, C. Qian, R. He, and C. C. Loy, "Everybody's talkin': Let me talk as you want," *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 585–598, 2022.

[15] X. Wen, M. Wang, C. Richardt, Z.-Y. Chen, and S.-M. Hu, "Photorealistic audio-driven video portraits," *IEEE Trans. Vis. Comput. Graphics*, vol. 26, no. 12, pp. 3457–3466, Dec. 2020.

[16] 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.

[17] Y. Fan, Z. Lin, J. Saito, W. Wang, and T. Komura, "Faceformer: Speech-driven 3D facial animation with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 18 770–18 780.

[18] Y. Chai, Y. Weng, L. Wang, and K. Zhou, "Speech-driven facial animation with spectral gathering and temporal attention," *Front. Comput. Sci.*, vol. 16, no. 3, pp. 1–10, 2022.

[19] A. Richard, M. Zollhöfer, Y. Wen, F. De la Torre, and Y. Sheikh, "Meshtalk: 3D face animation from speech using cross-modality disentanglement," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 1173–1182.

[20] T. Shimba, R. Sakurai, H. Yamazoe, and J.-H. Lee, "Talking heads synthesis from audio with deep neural networks," in *Proc. IEEE/SICE Int. Symp. Syst. Integration*, 2015, pp. 100–105.

[21] H. X. Pham, S. Cheung, and V. Pavlovic, "Speech-driven 3D facial animation with implicit emotional awareness: A deep learning approach," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 2328–2336.

[22] H. X. Pham, Y. Wang, and V. Pavlovic, "End-to-end learning for 3D facial animation from speech," in *Proc. 20th ACM Int. Conf. Multimodal Interaction*, 2018, pp. 361–365.

[23] Y. Zhou, Z. Xu, C. Landreth, E. Kalogerakis, S. Maji, and K. Singh, "VisemeNet: Audio-driven animator-centric speech animation," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–10, Jul. 2018.

[24] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[25] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 35–51.

[26] H.-Y. Lee et al., "DRIT++: Diverse image-to-image translation via disentangled representations," *Int. J. Comput. Vis.*, vol. 128, no. 10, pp. 2402–2417, 2020.

[27] F. Han, S. Ye, M. He, M. Chai, and J. Liao, "Exemplar-based 3D portrait stylization," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2021.3114308.

[28] H. Kim et al., "Neural style-preserving visual dubbing," *ACM Trans. Graph.*, vol. 38, no. 6, pp. 1–13, 2019.

[29] H. Zhu, H. Huang, Y. Li, A. Zheng, and R. He, "Arbitrary talking face generation via attentional audio-visual coherence learning," in *Proc. 29th Int. Joint Conf. Artif. Intell.*, 2021, pp. 2362–2368.

[30] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang, "Talking face generation by adversarially disentangled audio-visual representation," in *Proc. 33th AAAI Conf. Artif. Intell.*, 2019, pp. 9299–9306.

[31] Y. Zhou, X. Han, E. Shechtman, J. Echevarria, E. Kalogerakis, and D. Li, "Makelttalk: Speaker-aware talking-head animation," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–15, Nov. 2020.

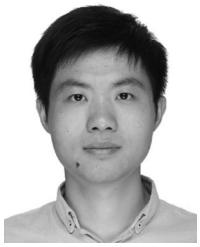
[32] G. Mittal and B. Wang, "Animating face using disentangled audio representations," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 3290–3298.

[33] X. Ji et al., "Audio-driven emotional video portraits," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14 080–14 089.

[34] S. Yao, R. Zhong, Y. Yan, G. Zhai, and X. Yang, "DFA-NeRF: Personalized talking head generation via disentangled face attributes neural rendering," 2022, *arXiv:2201.00791*.

[35] Y. Feng, H. Feng, M. J. Black, and T. Bolkart, "Learning an animatable detailed 3D face model from in-the-wild images," *ACM Trans. Graph.*, vol. 40, no. 4, pp. 1–13, Jul. 2021.
[36] DeepSpeech (0.1.0), Mozilla. Accessed: Mar. 25, 2022. [Online]. Available: https://github.com/mozilla/DeepSpeech
[37] D. Websdale, S. Taylor, and B. Milner, "The effect of real-time constraints on automatic speech animation," in *Proc. Interspeech*, 2018, pp. 2479–2483.
[38] J.-L. Schwartz and C. Savariaux, "No, there is no 150 ms lead of visual speech on auditory speech, but a range of audiovisual asynchronies varying from small audio lead to large audio lag," *PLoS Comput. Biol.*, vol. 10, no. 7, 2014, Art. no. e1003743.
[39] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[40] S. Gong, L. Chen, M. Bronstein, and S. Zafeiriou, "SpiralNet++: A fast and highly efficient mesh convolution operator," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 4141–4148.
[41] P. Senin, "Dynamic time warping algorithm review," *Inf. Comput. Sci. Dept. Univ. Hawaii Manoa Honolulu, USA*, vol. 855, no. 40, pp. 1–23, 2008.
[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–15.
[43] J. Thies, M. Zollhöfer, and M. Nießner, "Deferred neural rendering: Image synthesis using neural textures," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–12, Jul. 2019.
[44] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8798–8807.
[45] T. Unterthiner, S. van Steenkiste, K. Kurach, R. Marinier, M. Michalski, and S. Gelly, "FVD: A new metric for video generation," in *Proc. Int. Conf. Learn. Representations Deep Generative Models Highly Struct. Data*, 2019, pp. 1–9.
[46] C. Zhang et al., "3D talking face with personalized pose dynamics," *IEEE Trans. Vis. Comput. Graphics*, to be published, doi: 10.1109/TVCG.2021.3117484.
[47] R. Zheng, B. Song, and C. Ji, "Learning pose-adaptive lip sync with cascaded temporal convolutional network," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 4255–4259.
[48] 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.

Yujin Chai is working toward the PhD degree with the State Key Laboratory of CAD&CG, Zhejiang University. His research interests include deep learning and facial animation.



Tianjia Shao received the BS degree from the Department of Automation, Tsinghua University and the PhD degree in computer science from Institute for Advanced Study, Tsinghua University. He is a professor with the State Key Laboratory of CAD&CG, Zhejiang University. Previously, he was an assistant professor with the School of Computing, University of Leeds, UK. His research interests include 3D scene/object modeling, digital avatar creation, structure/function aware geometry processing, computer animation, and 3D printing.

Yanlin Weng received the bachelor's and master's degrees in control science and engineering from Zhejiang University, and the PhD degree in computer science from the University of Wisconsin - Milwaukee. She is currently an associate professor with the School of Computer Science and Technology, Zhejiang University. Her research interests include computer graphics and multimedia.

Kun Zhou (Fellow, IEEE) received the BS and PhD degrees in computer science from Zhejiang University in 1997 and 2002, respectively. He is a Cheung Kong professor with the Computer Science Department of Zhejiang University, and the director with the State Key Lab of CAD&CG. Prior to joining Zhejiang University in 2008, he was a leader researcher of the Internet Graphics Group with Microsoft Research Asia. His research interests are in visual computing, parallel computing, human computer interaction, and virtual reality. He is a fellow of ACM.