Audio-Driven Talking Face Video Generation with Dynamic Convolution Kernels

Zipeng Ye, Mengfei Xia, Ran Yi, Juyong Zhang, *Member, IEEE*, Yu-Kun Lai, *Member, IEEE*, Xuwei Huang, Guoxin Zhang, Yong-Jin Liu, *Senior Member, IEEE*

***Abstract*—In this paper, we present a dynamic convolution kernel (DCK) strategy for convolutional neural networks. Using a fully convolutional network with the proposed DCKs, high-quality talking-face video can be generated from multi-modal sources (i.e., unmatched audio and video) in real time, and our trained model is robust to different identities, head postures, and input audios. Our proposed DCKs are specially designed for audio-driven talking face video generation, leading to a simple yet effective end-to-end system. We also provide a theoretical analysis to interpret why DCKs work. Experimental results show that our method can generate high-quality talking-face video with background at 60 fps. Comparison and evaluation between our method and the state-of-the-art methods demonstrate the superiority of our method.**

***Index Terms*—dynamic kernel, convolutional neural network, multi-modal generation task, audio-driven talking-face generation.**

I. Introduction

TALKING-FACE video refers to video which mainly focuses on head or upper body of the speaker given audio or text signals. It has wide range of applications in news, TV shows, commercials, online chat, online courses, etc. According to the types of input signals, there are text-driven (e.g., [1]), audio-driven (e.g, [2]–[11]) and video-driven (e.g., [6], [12]–[17]) talking-face systems. In this paper, we propose an audio-driven talking-face system, capable of transferring the input talking-face video to a generated one corresponding to the input audio. It is a naturally cross-modal task with video and audio (i.e., of visual and auditory modalities) as input. The two modalities are strongly correlated [18], and thus it is possible to drive the talking-face video using an audio.

In this paper, we consider multi-modal fusion in a generation task, i.e., audio-driven talking-face video generation. For this task, a direct way is to treat multi-modal input as different features. To align these features, we can rearrange the audio features as additional channels of image frames and concatenate them with the image features. However, this method maps the audio feature elements to fixed locations, and as we will later show in our experiments presented in Section VI-D, it only works under special conditions where all the frames are aligned (i.e., each frame containing a frontal face at a fixed position), which are difficult to meet in practice. Another possible way is to use landmark points or parametric models [19], [20] as a prior, which can be inferred from the audio sequence. Facial landmarks are highly correlated to expression but also sensitive to head pose, view angle and scale. Therefore, it is necessary to align input photo/frames with a standard face, which has challenges dealing with the following: (1) facial image fusion with background, (2) head motion and (3) extreme head pose. 3D parametric models can be used as a strict and precise prior, which preserves almost all the information of expression and lip motion, and we can render an image using the parametric model. However, parametric models only contain low frequency information and the rendered images are often not photo-realistic. Therefore, post-processing is needed, which makes the pipeline complex and time-consuming. On the other hand, using these priors, it is difficult to design an end-to-end system with a fully convolutional neural network (FCNN), which is desired to ensure generalizability.

To overcome these drawbacks, in this paper, we propose a novel dynamic convolution kernel (DCK) technique that works well with FCNN for multi-modal generation tasks. Our key idea is to use a network to infer DCKs in a FCNN from audio modality. Then this FCNN can work with diverse input videos that have different head poses. Our model, i.e., FCNN with DCKs, is a network with dynamic parameters. In the literature, a few dynamic CNN parameter methods existed [21]–[25]. However, they were all proposed for processing single modal information and due to limited adaptivity, they are difficult to be extended to handle cross-modal applications. See Section II-B for more details. Our DCKs are totally different in both purpose and content: (1) DCKs are for multi-modal tasks, where the kernel is inferred from input audio, and (2) DCKs use completely flexible kernels, and are linear once the kernels are determined.

In this paper, we consider the following characteristics in our audio-driven talking-face system: (1) *Real time:* the video can be generated online when the audio signal is available; (2) *High quality:* the quality of generated video should be good enough such that people cannot easily distinguish between real video and generated video; (3) *Identity preserving:* the identity of the generated video should be preserved with the input video or photo; (4) *Expression and voice synchronization:* the expression and lip motion of the generated video should be

Z. Ye, M. Xia, Y.-J. Liu are with MOE-Key Laboratory of Pervasive Computing, the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

R. Yi is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University.

J. Zhang is with the School of Mathematical Sciences, University of Science and Technology of China.

Y.-K. Lai is with School of Computer Science and Informatics, Cardiff University, UK.

X. Huang and G. Zhang is with Kwai Inc.

Y.-J. Liu and R. Yi are corresponding authors. E-mail: liuyongjin@tsinghua.edu.cn.

synchronized with the input audio; (5) *Head motion:* the head pose and head motion of the generated video should be natural.

To address these characteristics, we propose DCKs and use them to build an end-to-end and one-for-all system, which only needs to be trained once and can work for different identities. To make better use of the multi-modal inputs which are difficult to fuse, we design DCKs which are different from traditional static convolution kernels. Once the model is trained, traditional convolution kernels no longer change. In contrast, our DCK will change with different inputs. We use the pre-trained audio network [26] to extract audio features and train a fully connected network to infer the DCKs from the input audio, and therefore we can design a fully convolutional network for video with different audio inputs well handled. We adapt the U-net [27] for DCKs by replacing convolutional kernels at selected layers to DCKs. Furthermore, we propose a novel dataset (including real videos and synthetic videos) to train our model in a supervised way.

In summary, the main technical contributions of our work include:

- We propose DCKs as an effective way to generate high-quality talking-face video from multi-modal input in *real time* with background and natural head motion, which is simple yet effective.
- We provide a theoretical analysis to explain DCKs' effectiveness.
- We propose a novel mixed dataset, including both real videos and synthetic videos, to supervise the training of our model.

II. Related Work

A. Multi-modal Fusion

One key challenge in tasks with multi-modal input is how to effectively fuse features in them. In various engineering fields, many algorithms have been proposed for fusing features collected from different types of sensors, which may have different modalities, rates, formats or confidence levels. The Kalman filter [28] is a classical algorithm for multi-sensor fusion. Bayesian inference [29] is another classic technique to fuse different features. For full details of existing fusion methods, the reader is referred to recent surveys [30], [31] and references therein.

Our study focuses on the neural network techniques. In this domain, a simple way for feature fusion is to directly concatenate features. The other simple way is to use different networks for extracting features of different modalities and use late feature fusion. The two simple strategies work well in classification and regression tasks, and achieve successes in many applications (e.g., [32]–[34]). Video is the most common input with different modalities. For classification and regression tasks of video, some learning-based methods [35]–[38] are proposed, which design network structures for fusing multi-modal input. On the other hand, the talking face video generation is a generation task, which is quite different from classification and regression tasks, and the simple concatenation strategy often fails. The methods which are designed for classification and regression tasks also fail. In some image generation tasks, using landmarks and 3D models as priors is useful to fuse multi-modal input (e.g., [2], [9]–[11], [39]). However, as we mention in Section I, it is difficult to design a fully CNN and an end-to-end system using 2D or 3D prior. In this paper, we propose a novel fully convolutional network with DCKs for the talking face video generation task with multi-modal input.

B. Neural Networks with Dynamic Parameters

The new model proposed in this paper, i.e., the fully convolutional network with DCKs, is a neural network with dynamic parameters. In recent years, several research works on designing neural networks with dynamic parameters have been proposed. The HyperNetworks [21] uses a hypernetwork to generate the weights for the other network, which has the similar idea as ours, but their motivations (for language modeling) and network structures (using recurrent neural network) are completely different from ours.

For CNNs, although fixed kernels are dominant in most research, there exist adaptions of CNNs ([22]–[25]) whose kernels can be dynamically adjusted. However, all these methods can only handle single mode information as input. The work [22] is designed for a classification task, which estimates a set of weights from input and these weights are used for balancing the output of nine sub-network-structures. Although the weights can be dynamically set, only adjusting the weights of nine sub-structures has limited capacity (that is suitable for single mode input); while our DCKs can adaptively set up to 10^4 parameters, which are more powerful and suitable for multi-modal input. The work [25] is similar to [22] in the spirit of using dynamic weights to adjust the linear combinations of a few convolution layers. The works [23], [24] predict a feature from input (different input may lead to different feature) and use this feature to do convolution at fixed positions in the network. As a comparison, our DCKs can predict parameters in different positions in the network and thus are more flexible.

C. Audio-driven Talking-face Video Generation

Audio-driven talking-face video generation is a task that uses an audio to drive a specified face (from a face photo or a talking face video) and produce a talking face video, with focus on fine talking details in head or upper body of the speaker [2]–[11], [39]. It is a typical task that uses multi-modal input. Many previous works use facial landmarks or 3D morphable models (3DMM) [19] as priors to bridge the two modalities. Inferring the prior from audio and using it to generate talking face video have been used in some practical methods [2], [9]–[11], [39]. However, it is difficult to use 2D or 3D prior to design an end-to-end system which can address the five characteristics summarized in Section I.

In [4], [5] two end-to-end methods are proposed that do not use 2D or 3D priors. The work [5] generates talking face video by the composition of a subject-related part and a speech-related part. Based on this composition, they propose an encoder-decoder model to disentangle the two parts from the video and use the input audio as the speech-related part to generate video. The work [4] proposes a conditional recurrent

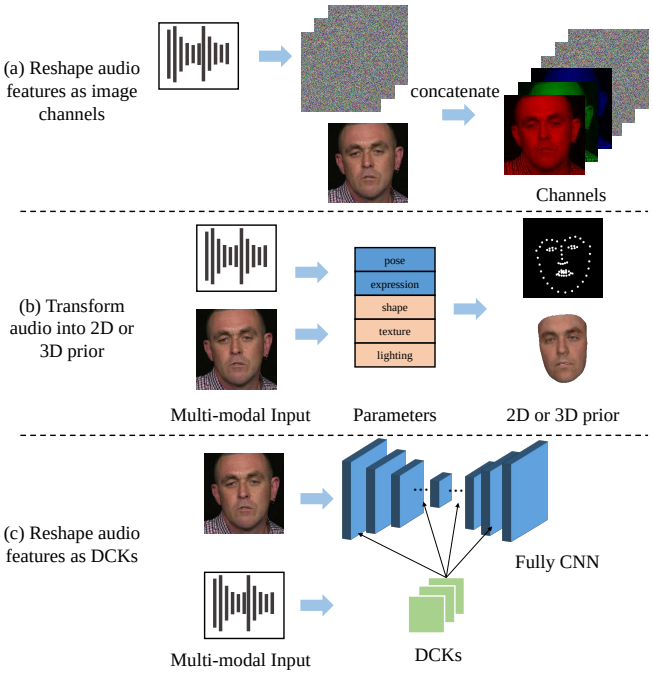

Fig. 1. Three strategies for fusing multi-modal input. (a) A simple and direct strategy is to extract the features of different modes, and concatenate them to feed into a network [3], [5]. (b) The second strategy is to use 2D [2] or 3D [9]–[11], [39] prior. It infers the prior's parameters from the audio input. (c) We propose a novel strategy, which extracts features from audio input and reshape features as DCKs of fully convolutional network.

generation network which uses an audio and an image as input, and output a video. Both methods can only work with a fixed standard head pose and output a video without head motion. As a comparison, our method can work with different poses and generate natural head motion.

III. Dynamic Convolution Kernels (DCKs)

A. Motivation

In this paper, we deal with a talking-face video generation task whose input is a pair of unmatched audio and video. The input contains entirely different modalities which have different forms and contents. How to fuse them together to effectively guide the training process is not easy and many works have studied this. As discussed in Section I, our target is to generate high-quality video with head motion. How to design the fusion is key to achieving these targets. Currently, there are two popular strategies to perform the fusion:

The first strategy for multi-modal input is to extract their features, and concatenate them or input them together to a network [3], [5] (Figure 1a). For example, we can use encoders to transform them into vectors and use a decoder to generate a video, which is not a fully convolutional network. We can also reshape the features of the input audio as channels of an image and concatenate them with the image, which works in a special case that the images are aligned so each pixel position has a fixed content. However, it is hard to achieve success in general because images are usually not aligned strictly and there is no fixed semanteme at each pixel. Our experimental results in Section VI-D demonstrate this observation.

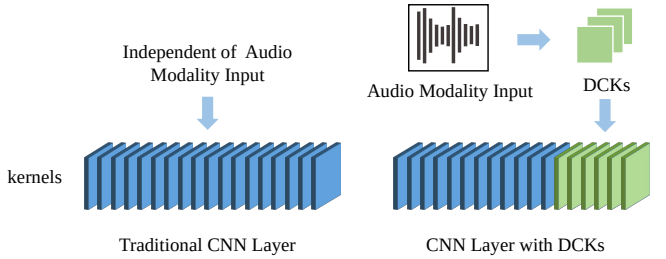

Fig. 2. Illustration of a CNN layer with DCKs. The blue are static convolution kernels and the green are dynamic convolution kernels. We reshape the features of audio modality into the shape of convolution kernels and use them to complete the CNN layer.

The second strategy is to use a parametric model as 3D prior (e.g., [9]–[11], [39]) or use facial landmarks as 2D prior (e.g., [2]). As shown in Figure 1b, they use audio to predict the prior and then use the prior to conduct the generation of videos. Using facial landmarks requires alignment of images and using 3D prior usually leads to high time cost, as demonstrated by our experimental results in Section VI-E.

To directly output high-quality video frames, a fully convolutional network is usually preferred to ensure generalizability. In this work, we design dynamic convolution kernels (DCKs), which are different from traditional static convolution kernels (Figure 1c). Once the model is trained, traditional convolution kernels no longer change. In contrast, our DCK is designed to infer from different inputs and therefore can change during the inference process (Figure 2). We use the convolution kernel as part of the generative network, which is dynamic for different input audios.

B. The Structure of DCKs

A traditional fully convolutional network $f(x)$, whose convolution kernels are $K(f) = \{k_1, k_2, \ldots, k_n\}$, is a transformation that iteratively applies the convolution operation to the input x. Denote by $y_0, y_1, \ldots, y_n$ the intermediate results where $x = y_0$ is the input and y_n is the output, we have:

$$y_i = g_i(k_i^\star y_{i-1}) \quad for \quad i = 1, 2, \ldots, n, \tag{1}$$

where $k_i^\star$ is the the ith convolution operator whose kernel is k_i and g_i is the ith combination of normalization and activation function. In this case, all convolution kernels are static which are learned from the training set and will not change in inference.

We propose a fully convolutional network using DCKs. Some selected convolutional layers $K_d \subset K(f)$ are no longer fixed after training, but instead are determined based on the input audio via a neural network, i.e. $k_j = h_j(A)$, for $k_j \in K_d$, where h_j is a neural network to determine the jth DCK given the input audio A. More generally, denote by $\Theta(f)$ all the parameters of a fully convolutional network, i.e. parameters of all convolution kernels, and some selected parameters $\Theta_d(f) \subset \Theta(f)$ are no longer fixed after training. A traditional fully convolutional network $f(x)$ can be written as $f(x; \Theta(f))$. The corresponding network with DCKs is $f(x; \Theta_s(f), \Theta_d(f))$, where $\Theta_s(f) = \Theta(f) \setminus \Theta_d(f)$ is the rest

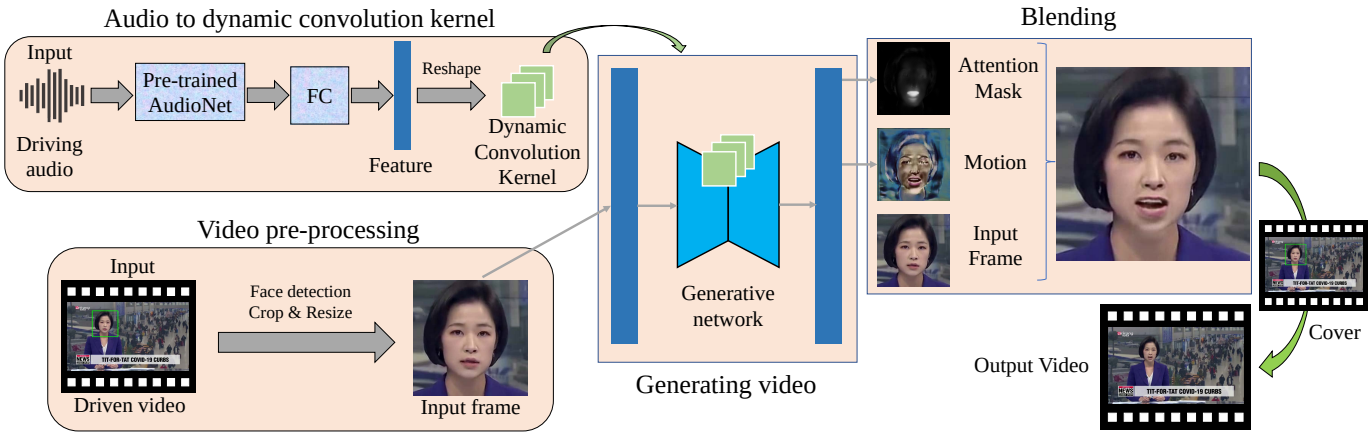

Fig. 3. The pipeline of our system and the architecture of our model. We adapt the U-net by incorporating dynamic convolution kernels as our generative network. We use the pre-trained audio network [26] to extract audio features and train a fully connected network to infer the dynamic convolution kernels from the input audio and use them to replace some traditional static convolution kernels. We detect the facial area from input video and crop it as the input of the generative network. The outputs of the generative network are an attention mask and a motion image. We blend them with the input to obtain the generation result.

static convolution kernels. We use a network to dynamically generate $\Theta_d(f)$ based on features of the input audio. We (1) use the pre-trained audio network in Wav2Lip [26], which consists of 2D convolutional layers and residual blocks, to extract audio features from Mel Spectrogram of input audio and (2) train a fully connected network to infer $\Theta_d(f)$ from the audio features. We have:

$$\Theta_d(f) = h_2(h_1(A)), \tag{2}$$

where A is the Mel Spectrogram of input audio, h_1 is the pre-trained audio network [26] and h_2 is the fully connected network. We reshape the output of the audio network into the shape of convolution kernels and use them to complete the fully convolutional network. Therefore, the fully convolutional network is dynamic with different input audios.

The advantages of using DCKs include the following aspects: (1) We can design a fully convolutional network for the input video, leading to real-time performance; (2) The convolution kernel is dynamic and can effectively fuse features from multi-modal inputs; (3) There is no binding between features and positions of pixels so it can work in different poses and different translations.

We present a theoretical interpretation for DCKs in Section V, to explain why it is useful for the cross-modal talking face video generation task.

IV. The System

We propose an audio-driven talking face video generation system, whose inputs are unmatched audio and video, and the output is a synthetic video. The pipeline of our system is shown in Figure 3. It is an end-to-end approach by directly outputting the synthetic video without intermediate results. Our system can generate high quality results in real time by efficiently incorporating the DCK technique and a supervised training scheme.

A. Fully Convolutional Network with DCKs

Our system deals with a multi-modal generation task whose input includes both audio and video. We use the pre-trained audio network in Wav2Lip [26] to extract audio features from Mel Spectrogram of input audio, and train (1) a fully connected network to generate DCKs from audio features, and (2) a fully convolutional network with DCKs. For training the two networks, we propose a novel method to train our model in a supervised way.

B. Training

In our task, the inputs are a pair of unmatched video and audio, and the output is a synthetic video. Denote by $\mathcal{V}$ the space of talking-face video and $\mathcal{A}$ the space of audio of talking-face video. An audio-driven talking-face system is a function $f : \mathcal{V} \times \mathcal{A} \to \mathcal{V}$. For any $A \in \mathcal{A}$ and $V \in \mathcal{V}$, $f(V, A)$ is a synthetic video, which have the same identity as V and the same expression (including lip motion) as A.

We use a supervised training scheme to train our model. Ideally, we need a training set consisting of pairs of talking-face videos which have different lip motions and the same other attributes (including identity and head motion) to train our model. However, it is difficult to obtain this kind of training dataset of real videos because the condition is too strict: even in a real talking face video without head movement, it is hard to extract two frames with exactly the same head pose. We take an alternative approach that synthesizes videos and pairs them with real videos to build this kind of dataset. Some talking face generation methods [26] can generate a talking-face video from a reference video and an audio, where the generated video has the same identity and head motion as the reference video. We use the method [26] to generate a new training dataset by the following steps: (1) we collect $N_r = 550$ real talking-face videos $\{V_1^0, V_2^0, \ldots, V_{N_r}^0\}$ from video websites and collect $m = 550$ audios from talking-face videos, whose lengths are about 60 seconds; (2) for each real video $V_i^0, i = 1, 2, \ldots, N_r$, we use the method [26] to generate

m videos $V_i^1, V_i^2, \ldots, V_i^m$ which has the same identity and head motion with the video V_i^0 and different lip motions; (3) we combine V_i^0 with V_i^j as a pair of videos and then we have mN_r pairs of videos.

We use the dataset obtained above (including real videos and synthetic videos) to train our model. In the training set, all talking-face videos have their corresponding audios and we denote their relation by an operator $A(V) : \mathcal{V} \to \mathcal{A}$ which maps a talking-face video V to its corresponding audio A. For each batch, we randomly select a pair of videos V_1, V_2 and their corresponding audios $A_1 = A(V_1), A_2 = A(V_2)$. We use our model to generate video $f(V_1, A_2)$ from V_1 and A_2 and $f(V_2, A_1)$ from V_2 and A_1.

Reconstruction Loss. We consider $f(V_1, A_2), f(V_2, A_1)$ are generation results and V_2, V_1 are their ground truth. Ideally, the generation results should be exactly the same as the ground truth. We use reconstruction loss to constrain our model to generate talking face videos similar to the ground truth. The loss term is calculated as the L_1 norm of the difference between generation results and the ground truth, i.e.,

$$\begin{aligned} &L_{rec}(f) \\ &= \mathbb{E}_{V_1,V_2\sim\mathcal{V}}(\|f(V_1, A_2) - V_2\|_1 + \|f(V_2, A_1) - V_1\|_1). \end{aligned} \tag{3}$$

Adversarial Loss. We use adversarial loss to ensure that $f(\mathcal{V}, \mathcal{A})$ has the same distribution as $\mathcal{V}$, which can improve the quality of generation results. We adapt the adversarial loss of LSGAN [40] as:

$$\begin{aligned} &L_{adv}(f, D) \\ &= \mathbb{E}_{V_1,V_2\sim\mathcal{V}}(\|D(f(V_1, A_2))\|_2^2) + \mathbb{E}_{V\sim\mathcal{V}}(\|1 - D(V)\|_2^2). \end{aligned} \tag{4}$$

The overall loss function is in the following form:

$$L_{total}(f, D) = L_{adv}(f, D) + \lambda_{rec} L_{rec}(f), \tag{5}$$

where λ_{rec} is the weight for balancing the multiple objectives. For all experiments, we set $\lambda_{rec} = 10$. The optimization target is:

$$\min_f \max_D L_{total}(f, D). \tag{6}$$

C. Blending

Instead of directly generating the synthetic frames, the output of our method (shown in Fig. 4) is an attention mask α which is a grayscale image, and a motion image M which is a color image that presents the change. α determines at each pixel how much the output should be influenced by the motion image M. Denote by I the input image and I' the synthetic image, we have:

$$I' = I \otimes (1 - \alpha) + M \otimes \alpha, \tag{7}$$

where $\otimes$ is pixel-wise multiplication.

Compared with directly generating the synthetic frames, our method has the following advantages. It can not only enforce the network to focus on audiovisual-correlated regions but also offers an efficient post-processing to produce desired output, avoiding expensive image fusion. It also increases the interpretability of the network and we can be informed of where the network focuses. In practice, it is difficult to train the network with DCKs by directly generating the synthetic frames. In Figure 9, the generation results of directly generating the synthetic frames have different skin color from inputs whereas those of blending have the same skin color as inputs.

D. Adding Background in Real Time

Our system does not need to use image fusion to integrate the generated results with the background, which usually takes a long time. We directly cover the generated results onto the background instead of image fusion and this helps save time. The boundaries of generation results are the same as those of inputs, because we do not change the head pose and non-face area. Due to the attention mechanism and the mask loss, the direct covering has good performance and there are no visible artifacts on the boundary between the face and background. A frame with background generated by direct covering is shown in Figure 5.

As a comparison, other state-of-the-art methods (e.g., [2], [3], [5], [6], [9]–[11], [26], [39], [41]) either cannot keep the head pose or change the non-face area, and therefore, the face and background have different color values on the boundary. We note that in these methods, directly covering would cause inconsistency between the generated region and background. Therefore, they cannot adopt our directly covering scheme as efficient as ours; e.g., Wav2Lip [26] also generates the facial region and uses the direct covering to add background, but as we will show in Section VI-E, their results exhibit clear boundaries between the generated region and background.

V. Theoretical Interpretation for DCKs

We can understand the dynamic convolution kernels (DCKs) in the following way. Denote by $\mathcal{T}$ the space of tasks such as all expressions. For a fixed task $t \in \mathcal{T}$ such as smiling, we can train a network to transfer an image to a new image with smile. For different tasks in $\mathcal{T}$, we can train different networks. We believe these networks are mostly the same with slight distinction. We use static convolution kernels to learn the common characteristics and use dynamic convolution kernels to learn the distinction. Therefore, the network with dynamic kernels can handle all tasks in the space $\mathcal{T}$.

We present the following formulations to provide a theoretical interpretation of the above statement. Some experimental validations are presented in Section VI-B.

A. Interpretation from set approximation

Assumption 1: A multi-modal task can be fulfilled by solving a set $\mathcal{T} = \{t_i\}$ of simpler tasks, each of which t_i can be fulfilled by a fully convolutional network f^i with fixed parameters. All the networks $\{f^i\}$ have the same structure and most of their parameters are the same.

Now we show that a single fully convolutional network with DCKs can well approximate the set of networks $\{f^i\}$ with bounded output errors.

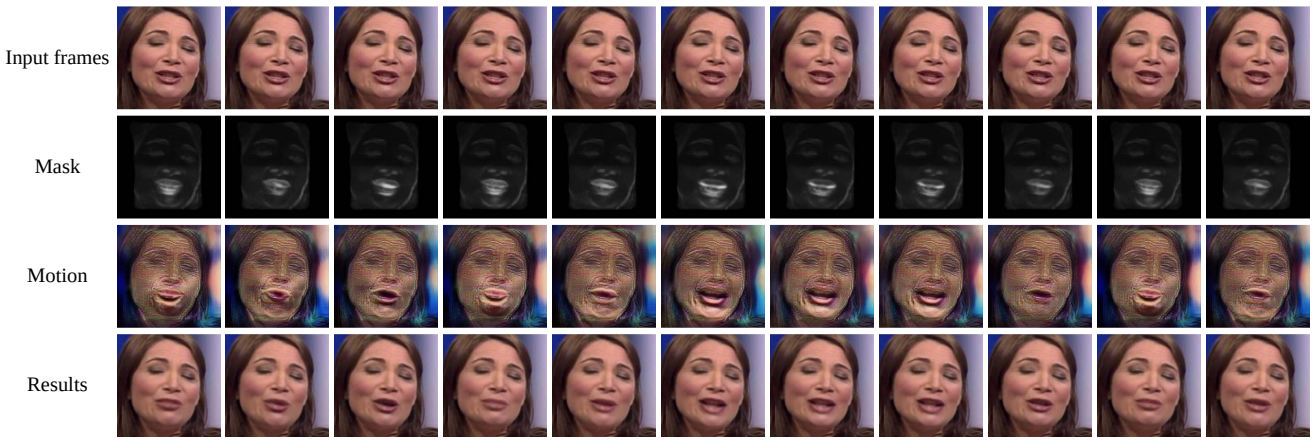

Fig. 4. Results of attention masks and motion images generated by our method. The audio input is 'for many of us', and the video input is obtained by repeating a face photo multiple times.

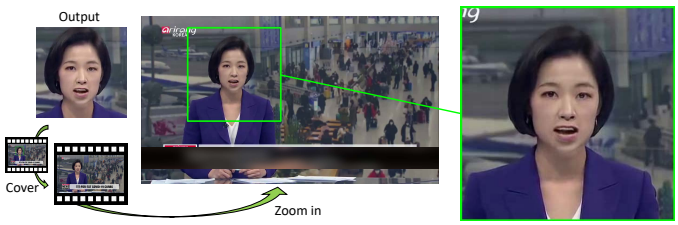

Fig. 5. Our system directly covers the generated results onto the background instead of image fusion. No visible artifacts are observed on the boundary between the face and background.

Lemma 1: Let the activation function g be any one of Leaky ReLU, $\tanh$ or Sigmoid. For any input $x^1, x^2 \in \mathbb{R}^n$, the following inequality holds:

$$\|g(x^1) - g(x^2)\|_p \leqslant \|x^1 - x^2\|_p. \tag{8}$$

Proof 1: For any $a \in \mathbb{R}$, $a - g(a)$ increases monotonically; hence for $a^1 \geqslant a^2, a^1, a^2 \in \mathbb{R}$, we have $a^1 - g(a^1) \geqslant a^2 - g(a^2)$, i.e., $|a^1 - a^2| = a^1 - a^2 \geqslant g(a^1) - g(a^2) = |g(a^1) - g(a^2)|$. That completes the proof.

Given two fully convolutional networks f^1 and f^2 corresponding to two sets of convolution kernels $\{k_1^1, k_2^1, \cdots, k_n^1\}$ and $\{k_1^2, k_2^2, \cdots, k_n^2\}$, respectively, where each pair of (k_i^1, k_i^2) has the same kernel size, let $\{y_0^1, \cdots, y_n^1\}$ and $\{y_0^2, \cdots, y_n^2\}$ be the two sets of intermediate results of two convolution networks f^1 and f^2, where $y_0^1 = y_0^2 = x$. Then we have:

$$y_i^j = g_i(k_i^{j*} y_{i-1}^j), \quad i = 1, 2, \cdots, n, \quad j = 1, 2, \tag{9}$$

where $*$ is the convolution operator and g_i is the activation function. The following theorem gives an upper bound of the difference between the two outputs $f^1(x), f^2(x)$ in terms of the L_p norm.

Theorem 1: If all convolution kernels have a uniform upper bound of their L_p norm, i.e., $\|k_i^j\|_p \leqslant M_p$ for $\forall i, j$ and some $M_p > 0$, the following inequality holds:

$$\|y_n^1 - y_n^2\|_p \leqslant M_p^{n-1} \|x\|_p \sum_{i=1}^{n} \|k_i^1 - k_i^2\|_p. \tag{10}$$

Proof 2: We prove this theorem by induction. First we consider the case $n = 1$, i.e., there is only one convolution kernel for each set. By calculating the L_p loss, we have

$$\begin{aligned} &\|y_1^1 - y_1^2\|_p = \|g(k_1^{1*} x) - g(k_1^{2*} x)\|_p \\ &\leqslant \|k_1^{1*} x - k_1^{2*} x\|_p = \|(k_1^1 - k_1^2)^* x\|_p \\ &\leqslant \|k_1^1 - k_1^2\|_p \cdot \|x\|_p, \end{aligned}$$

where the last inequality comes directly from the Cauchy inequality. Now, suppose the inequality (10) holds for $n \leqslant m-1$. For $n = m$, we have

$$\begin{aligned} &\|y_m^1 - y_m^2\|_p \\ =&\|g_m(k_m^{1*} y_{m-1}^1) - g_m(k_m^{2*} y_{m-1}^2)\|_p \\ \leqslant&\|(k_m^{1*} y_{m-1}^1 - k_m^{2*} y_{m-1}^2)\|_p \\ \leqslant&\|k_m^{1*} y_{m-1}^1 - k_m^{1*} y_{m-1}^2\|_p + \|k_m^{1*} y_{m-1}^2 - k_m^{2*} y_{m-1}^2\|_p \\ \leqslant&\|k_m^1\|_p \cdot \|y_{m-1}^1 - y_{m-1}^2\|_p + \|k_m^1 - k_m^2\|_p \cdot \|y_{m-1}^2\|_p \\ \leqslant&M_p \cdot M_p^{m-2} \|x\|_p \sum_{i=1}^{m-1} \|k_i^1 - k_i^2\|_p \\ &+ \|k_m^1 - k_m^2\|_p \|x\|_p \prod_{i=1}^{m-1} \|k_i^2\|_p \\ \leqslant&M_p^{m-1} \|x\|_p \sum_{i=1}^{m} \|k_i^1 - k_i^2\|_p \end{aligned}$$

That completes the proof.

In practice, the constant M_p is usually small, e.g., in all experiments in Section VI, $M_p = 0.4559$. Then Theorem 1 says that for two networks f^1 and f^2 with a fixed number n of convolution layers,

$$\|f^1(x) - f^2(x)\|_p \leqslant C_p \|x\|_p \sum_{i=1}^{n} \|k_i^1 - k_i^2\|_p, \tag{11}$$

where C_p is a constant independent of the input x and the convolution networks.

Note that although all the networks in the set $\{f^i\}$ have the same structure and most of their parameters are the same, the remaining parameters can be significantly different. Then any fully convolutional network with fixed parameters cannot

well approximate all the networks in $\{f^i\}$. Let $f \in \{f^i\}$ and f' be a fully convolutional network with DCKs which are inferred from the audio modality. If the inference makes the parameters in DCKs well approximate the parameters in the corresponding layers of f, f' can well approximate any f in $\{f^i\}$.

B. Interpretation from loss values

The value of the objective loss function can reflect the quality of generation results of the system to some extent. Next we present the error bounds of two loss terms in our objective function, showing the loss value is approximately optimal[1] using a network with DCKs. In addition, we also present the error bounds of cycle loss [42], which is a useful loss term.

Corollary 1: Let f_D^j, $j = 1, 2$, be the discriminator network, which is a fully convolutional network. Let $\{k_{n+1}^j, \cdots, k_{n+r}^j\}$ be the convolution kernels of the discriminator D^j (Here, the indexes follow those of the generator). The adversarial loss can be expressed as:

$$\begin{aligned}&L_{adv}(f^j, D^j)\\ =&\mathbb{E}_{x\sim\mathcal{X}}(\|D^j(f^j(x))\|_2^2) + \mathbb{E}_{x\sim\mathcal{X}}(\|1 - D^j(x)\|_2^2)\\ =&\mathbb{E}_{x\sim\mathcal{X}}(\|f_D^j \circ f^j(x)\|_2^2) + \mathbb{E}_{x\sim\mathcal{X}}(\|1 - f_D^j(x)\|_2^2).\end{aligned}$$

Then there exist constants $A_1, A_2 > 0$, such that the following inequality holds:

$$\begin{aligned}&|L_{adv}(f^1, D^1) - L_{adv}(f^2, D^2)|\\ \leqslant&\left(A_1 \left(DK_2\right)^2 + A_2 \cdot DK_2\right) \mathbb{E}_{x\sim\mathcal{X}}(\|x\|_2^2),\end{aligned}$$

where $DK_2 = \sum_{i=1}^{n+r} \|k_i^1 - k_i^2\|_2$.

Proof 3: We can regard the discriminator D^j as another convolution network together with convolution kernels $\{k_{n+1}^j, \cdots, k_{n+r}^j\}$ and activation functions $g_{n+1}, \cdots, g_{n+m}$. Notice that there exist $C_1, C_2 > 0$, such that the inequalities hold:

$$\|f_D^2 \circ f^2(x)\|_2 \leqslant C_1\|x\|_2, \text{ and}$$

$$\|1 - f_D^2(x)\|_2 \leqslant C_2\|x\|_2.$$

We denote

$$\begin{aligned}\|f_D^1 \circ f^1(x) - f_D^2 \circ f^2(x)\|_2 \leqslant \Delta_1\|x\|_2,\\ \|f_D^1(x) - f_D^2(x)\|_2 \leqslant \Delta_2\|x\|_2,\end{aligned}$$

where

$$\begin{aligned}\Delta_1 \propto \sum_{i=1}^{n+r} \|k_i^1 - k_i^2\|_2,\\ \Delta_2 \propto \sum_{i=1}^{r} \|k_{n+i}^1 - k_{n+i}^2\|_2.\end{aligned}$$

[1]Assume that the optimal loss value is provided by a network $f \in \{f^i\}$.

Then we have

$$\begin{aligned}&|L_{adv}(f^1, D^1) - L_{adv}(f^2, D^2)|\\ \leqslant&\mathbb{E}_{x\sim\mathcal{X}}(|\|f_D^1 \circ f^1(x)\|_2^2 - \|f_D^2 \circ f^2(x)\|_2^2|)\\ &+ \mathbb{E}_{x\sim\mathcal{X}}(|\|1 - f_D^1(x)\|_2^2 - \|1 - f_D^2(x)\|_2^2|)\\ \leqslant&\mathbb{E}_{x\sim\mathcal{X}}\left((\|f_D^2 \circ f^2(x)\|_2 + \Delta_1\|x\|_2)^2 - \|f_D^2 \circ f^2(x)\|_2^2\right)\\ &+ \mathbb{E}_{x\sim\mathcal{X}}\left((\|1 - f_D^2(x)\|_2 + \Delta_2\|x\|_2)^2 - \|1 - f_D^2(x)\|_2^2\right)\\ \leqslant&\mathbb{E}_{x\sim\mathcal{X}}(((C_1 + \Delta_1)^2 - C_1^2)\|x\|_2^2)\\ &+ \mathbb{E}_{x\sim\mathcal{X}}(((C_2 + \Delta_2)^2 - C_2^2)\|x\|_2^2)\\ =&2(C_1\Delta_1 + C_2\Delta_2)\mathbb{E}_{x\sim\mathcal{X}}(\|x\|_2^2)\\ &+ (\Delta_1^2 + \Delta_2^2)\mathbb{E}_{x\sim\mathcal{X}}(\|x\|_2^2)\\ \leqslant&\left(A_1 \left(DK_2\right)^2 + A_2 \cdot DK_2\right) \mathbb{E}_{x\sim\mathcal{X}}(\|x\|_2^2),\end{aligned}$$

where $A_1, A_2 > 0$ are two constants. That completes the proof.

Corollary 2: Let f_P be a fully convolutional network with kernels $\{k_1, \cdots, k_l\}$. The perceptual loss, including the VGG loss [43] and LPIPS loss [44], can be expressed as:

$$L_{pcpt}(f^j, f_P) = \mathbb{E}_{x\sim\mathcal{X}}(\|f_P(f^j(x)) - f_P(y)\|_1),$$

where y is the ground truth. Then there exist constants $C_{pcpt} > 0$ such that the following inequality holds:

$$\begin{aligned}&|L_{pcpt}(f^1, f_P) - L_{pcpt}(f^2, f_P)|\\ \leqslant&C_{pcpt}(\sum_{i=1}^{n} \|k_i^1 - k_i^2\|_1)E_{x\sim\mathcal{X}}(\|x\|_1).\end{aligned}$$

Specially, if f_P degenerates into an identity transformation, the perceptual loss degenerates into L_1 loss in Eq. 3.

Proof 4: It is a direct consequence of Theorem 1.

Corollary 3: Let $f_{\mathcal{X}\to\mathcal{Y}}^j$ and $f_{\mathcal{Y}\to\mathcal{X}}^j$, $j = 1, 2$, be the generators from the domain $\mathcal{X}$ to the domain $\mathcal{Y}$ and from $\mathcal{Y}$ to $\mathcal{X}$, respectively, which are fully convolutional networks. The cycle loss term [42] can be expressed by

$$L_{cyc}(f^j) = \mathbb{E}_{x\sim\mathcal{X}}(\|f_{\mathcal{Y}\to\mathcal{X}}^j(f_{\mathcal{X}\to\mathcal{Y}}^j(x))) - x\|_1). \tag{12}$$

Then by considering $f_{\mathcal{Y}\to\mathcal{X}}^j \circ f_{\mathcal{X}\to\mathcal{Y}}^j$ as a $2n$-layer fully convolutional network, we have:

$$\begin{aligned}&|L_{cyc}(f^1) - L_{cyc}(f^2)| \leqslant\\ &C_{cyc}\left(\sum_{i=1}^{n}(\|Dk_{i,\mathcal{X}\to\mathcal{Y}}\|_1 + \|Dk_{i,\mathcal{Y}\to\mathcal{X}}\|_1)\right)\mathbb{E}_{x\sim\mathcal{X}}(\|x\|_1),\end{aligned}$$

where C_{cyc} is a constant, $Dk_{i,\mathcal{X}\to\mathcal{Y}} = k_{i,\mathcal{X}\to\mathcal{Y}}^1 - k_{i,\mathcal{X}\to\mathcal{Y}}^2$ and $Dk_{i,\mathcal{Y}\to\mathcal{X}} = k_{i,\mathcal{Y}\to\mathcal{X}}^1 - k_{i,\mathcal{Y}\to\mathcal{X}}^2$.

Proof 5: We have

$$\begin{aligned}&|L_{cyc}(f^1) - L_{cyc}(f^2)|\\ \leqslant&\mathbb{E}_{x\sim\mathcal{X}}(\big|\|f_{\mathcal{Y}\to\mathcal{X}}^1 \circ f_{\mathcal{X}\to\mathcal{Y}}^1(x) - x\|_1\\ &- \|f_{\mathcal{Y}\to\mathcal{X}}^2 \circ f_{\mathcal{X}\to\mathcal{Y}}^2(x) - x\|_1\big|)\\ \leqslant&\mathbb{E}_{x\sim\mathcal{X}}(\|f_{\mathcal{Y}\to\mathcal{X}}^1 \circ f_{\mathcal{X}\to\mathcal{Y}}^1(x) - f_{\mathcal{Y}\to\mathcal{X}}^2 \circ f_{\mathcal{X}\to\mathcal{Y}}^2(x)\|_1)\\ \leqslant&C_{cyc}\left(\sum_{i=1}^{n}(\|Dk_{i,\mathcal{X}\to\mathcal{Y}}\|_1 + \|Dk_{i,\mathcal{Y}\to\mathcal{X}}\|_1)\right)\mathbb{E}_{x\sim\mathcal{X}}(\|x\|_1).\end{aligned}$$

That completes the proof.

Fig. 6. Given different audio inputs (successive 3 frames of the word 'for'), our system f' can infer different DCKs (i.e., DCKs#1,#2 and #3), such that f' with DCKs#1,#2 and #3 can transfer a face (with arbitrary expression in any input frame) into expressions with different mouth shapes.

VI. Experiments

A. Implementation Details

We implemented our method with PyTorch [45] and OpenCV. We trained the model on a server with an Intel Xeon Gold 6126 (2.60 GHz) and a NVIDIA TITAN RTX GPU. We also tested it on the same server. We use a mixed video dataset (including real videos and synthetic videos) described in Sec. IV-B to train our model.

Our system starts with video pre-processing (Figure 3). For a video with background, we crop the facial area (detected by Dlib [46]) from the video and resize it to 256×256 as the input of our system. At the end of the pipeline, we cover the facial area with generated results directly without image fusion, which is a major advantage of our method that helps achieve real-time performance.

We use a U-net with DCKs, called Adapted U-net, as the generator, which has 5 down-sampling layers, 4 middle layers and 5 up-sampling layers, where all middle layers are with DCKs. We use the pre-trained audio network [26], which consists of 2D convolutional layers and residual blocks, to extract audio features from Mel Spectrogram of input audio, where the parameters of Mel Spectrogram are the same as [26]. For each layer with a dynamic convolution kernel, we train a fully connected network to infer the DCK from the audio features. We reshape the output of this module to the shape $l \times c_1 \times (c_2 \times ks \times ks + 1)$, where l is the length of video sequence, ks is the kernel size, c_1 and c_2 are the numbers of channels of output and input of the corresponding convolution layers of the adapted U-net. In all our experiments, $ks = 1$, $c_1 = 256$ and $c_2 = 256$. We implement the DCKs by convolution operators with the group parameter in PyTorch.

B. Validation of DCK understanding

Our system is a fully convolutional network f' with DCKs. The theoretical interpretation in Section V indicates that our system f' can well approximate a set of networks $\{f^i\}$ with fixed parameters, such that according to different audio input, the system f' can adaptively choose the desired $f \in \{f^i\}$. Some results are shown in Figure 6. Given different audio inputs, our DCKs can successfully infer different parameters that drive the system f' to approximate different $f \in \{f^i\}$, e.g., transforming a face (with arbitrary expression in any input frame) into other expressions with different mouth shapes. Another advantage of the system f' with DCKs is that for any finite set of $\{f^i\}$, each input frame can only be transferred into other expressions in a finite expression space, while our DCKs can infer parameters in a continuous space, so that our system can provide better mechanism to secure the inter-frame continuity.

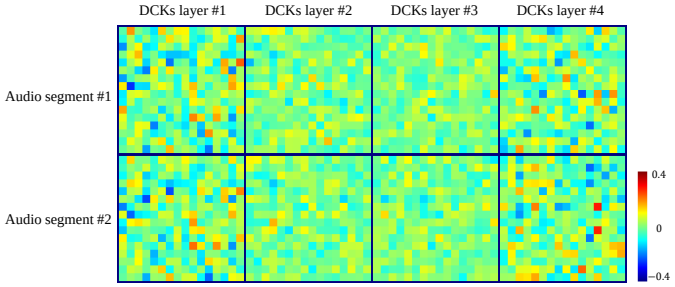

Fig. 7. Four DCKs layer inferred from two audio segments are showed. Different audio inputs lead different parameters of convolution kernel. For better observation, we only visualize part of parameters of DCKs (i.e., the first 16×16 parameters).

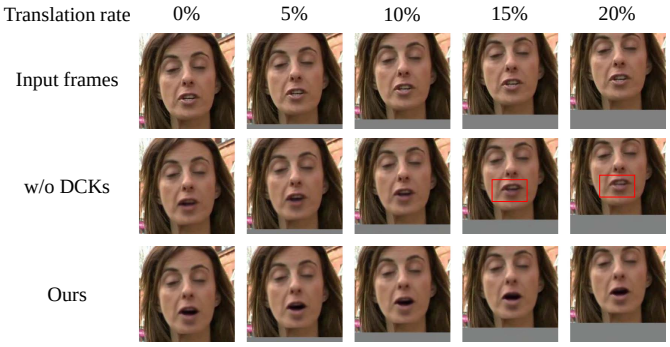

Fig. 8. Ablation study for comparing f_{w/o_DCKs} and our system f'. The phoneme of input audio is /ba:/ of the word 'Obama'. See text for details.

Fig. 9. Ablation study. The first row shows several frames of real videos as input frames. We use the same audio features as input audio for all the input frames. The syllable of input audio is the word 'were'. The second to the fourth rows show the generation results of our method with one of the modules disabled along with our full model. The generation results without DCKs sometimes have wrong lip motion. The generation results of directly generating the synthetic frames without blending are almost the same as input frames (mouth shape in particular), which shows our method without blending (i.e. the attention mechanism) is difficult to generate good talking-face videos. Only the whole method can generate good results in all cases.

C. Visualization of DCKs

A fully convolutional network with DCKs is a black box. Visualization of DCKs helps us understand it how to work. In all our experiment, kernel sizes of DCKs are 1 so we can visualize each DCKs layer inferred from an audio segment as an image whose weight and height are the numbers of input and output channels of the DCKs layer. The DCKs layer could be regarded as the correlation coefficient between different channels, i.e. their covariance matrix. Four DCKs layer inferred from two audio segments are showed in Figure 7. Results show different audio inputs lead different parameters of convolution kernel.

D. Ablation Study

The ablation study focuses on the novel DCKs and blending. First, we design a network f_{w/o_DCKs} without DCK, which reshapes the audio features to $16 \times H \times W$ (i.e., 16 channels and the same resolution $H \times W$ as the image), and use the spatial attention fusion module of the method [47] to fuse audio features and image features. As shown in Figure 8, we compare f_{w/o_DCKs} with our fully convolutional network f' with DCKs in the following test scenario: the face region in the input frame is moved upwards (defined by a translation rate which is the number of upwards translation pixels over the frame height). The results show that f' is invariant for translation and outputs similar results for different translation rates, while f_{w/o_DCKs} outputs different results for different translation rates, which leads to bad lip synchronization for slightly larger translation rate (Figure 8 middle row, 15% and 20%). The reason is possibly that reshaping the audio features to image space requires fixed semanteme at each pixel.

To demonstrate the effectiveness of our blending scheme, we train a network to directly generate frames without blending. The qualitative results are shown in Figure 9. We use the same audio features as input audio for all the input frames, whose syllable is the word 'were'. We compare the generation results of our method with and without the two modules, i.e. DCKs and blending. The generation results without DCKs sometimes have wrong lip motion. The generation results of directly generating the synthetic frames without blending are almost the same as input frames (mouth shape in particular), which shows our method without blending (i.e. the attention mechanism) is difficult to generate good talking-face videos. Only the whole method can generate good results in all cases.

TABLE I
QUANTITATIVE COMPARISON BETWEEN OURS AND ABLATION STUDY METHODS.

Metric	w/o Blending	w/o DCKs	Ours
PSNR ↑	29.29	31.08	**31.98**
SSIM ↑	0.74	0.78	**0.81**
LMD ↓	1.65	1.61	**1.44**

We use the test set of LRW dataet [48] for quantitative metric evaluation. For each video in the test set, we input its first frame and audio signal to the network, and generate a talking-face video for each comparison method. We compare the results with the ground-truth videos, after aligning them according to the way used in ATVGnet [2]. We use Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Metrics (SSIM) to evaluate the quality of images, and use Landmark Distance (LMD) [49] to evaluate the accuracy of lip movement. The results of quantitative comparison are summarized in Table I. We can see that DCKs and blending are helpful for generating talking-face videos.

E. Comparison with State of the Arts

In this section, we compare our model with state-of-the-art methods, including ATVGnet [2], You Said That [3], Wav2Lip [26], X2Face [6], DAVS [5], SDA [41], Yi's Method [9]. We first introduce and discuss these methods.

ATVGnet. ATVGnet [2] generates talking-face video in real time from input photo and audio by hierarchical networks. It crops the facial area from the input photo and aligns it by affine transformation based on facial landmarks extracted from Dlib [46]. Because the alignment operation changes and fixes the view angle, results of ATVGnet are talking-face videos without background and head motion. The resolution of its generation results is 128×128, which is lower than ours. Its results has neither head motion nor eye blink.

You Said That. You Said That [3] generates talking-face video from input photo and audio using two CNNs to extract features from audio (spectrum) and photo separately and then concatenating them in channels. It aligns the input photo by applying spatial registration so results of You Said That are talking-face videos without background, head motion and eye blink. The resolution of its generation results is 112×112, which is lower than ours.

Wav2Lip. Wav2Lip [26] generates talking-face video in real time from input photo (or input video) and audio using CNNs with encoder-decoder structure. It uses a lip-sync expert model to train the generation model in order to achieve high

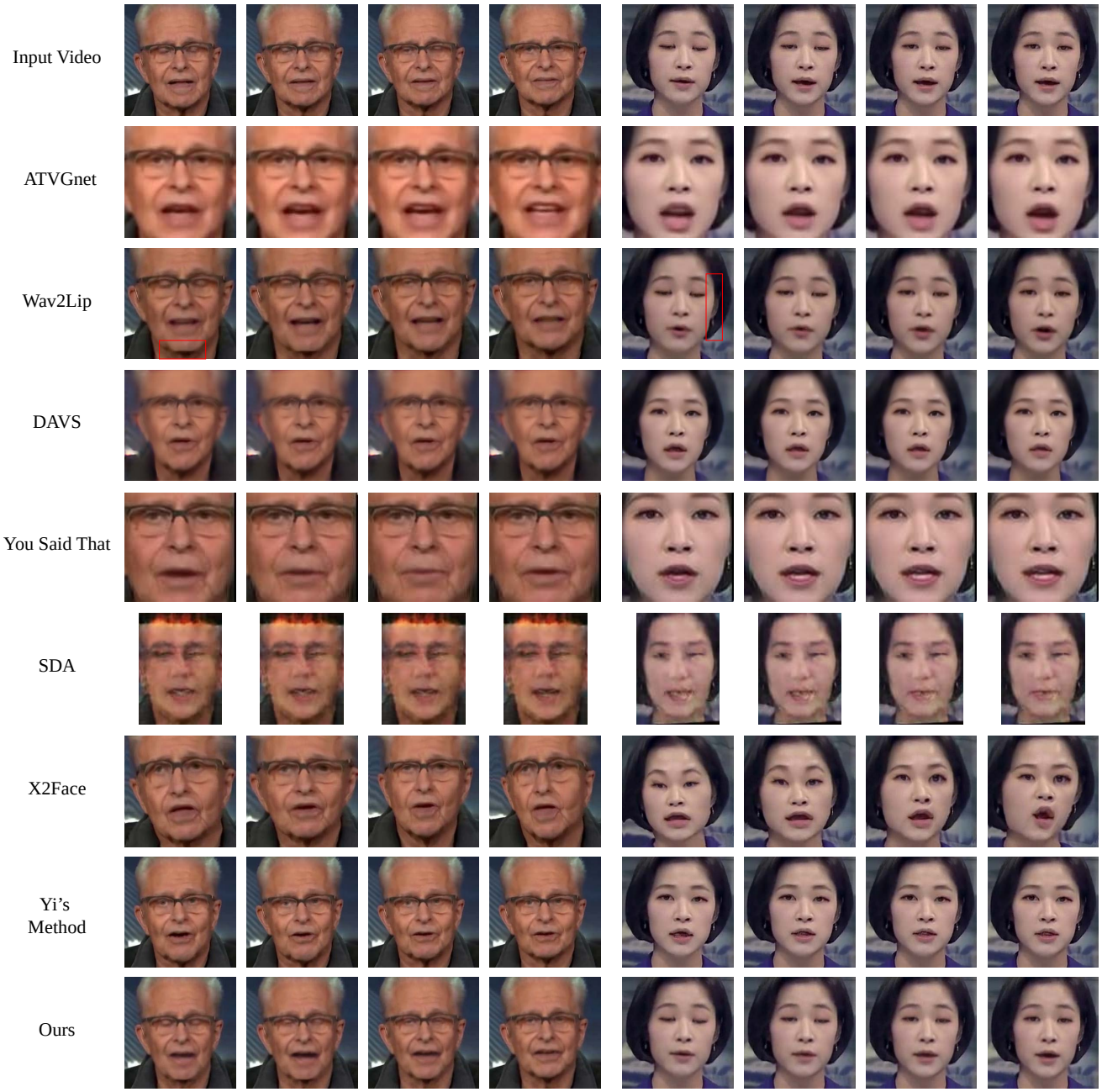

Fig. 10. The qualitative comparison between the STOA methods (Wav2Lip [26], X2Face [6], You said that [3], SDA [41], DAVS [5], ATVGnet [2] and Yi's Method [9]) and ours. Results of SDA are apparently worse than others. X2Face distorts the shape of face and changes the identity. ATVGnet, DAVS and You Said That results have no head motion and no eye blink. ATVGnet results are of low resolution and degraded visual quality. Wav2Lip results are of low definition in facial area, which sometimes causes an obvious boundary (red boxes) between facial part and other parts when directly covering the generated region onto the background. Yi's Method can not generate talking-head videos in real time. While our method can generate high-quality talking-head videos with head motion in real time. The syllables of input audios are the phrase 'I am' in the left and the word 'really' in the right.

lip synchronization. However, the resolution of its generation results is 96×96, which is lower than ours.

X2Face. X2Face [6] controls a source frame using another frame with different identities to produce a generated frame with the identity of the source frame but the pose and expression of the other frame. It uses an auto-encoder to edit a frame in hidden space so the generation process can be driven by audio. However, the model can not totally disentangle different attributes of multi-modal inputs in hidden space which leads to unstable and discontinuous generation results. It can not keep the head pose so results of X2Face are talking-face videos without background.

DAVS. DAVS [5] generates talking-face video from input photo and audio using an auto-encoder to disentangle subject-related information and speech-related information via an associative-and-adversarial training process. However, the model can not totally disentangle speech-related information in hidden space, which leads to low audio-visual consistency. It cannot keep the boundary of the cropped face so results of DAVS are talking-face videos without background. It also has

TABLE II
QUANTITATIVE COMPARISON BETWEEN OURS AND THE STOA METHODS (WAV2LIP [26], X2FACE [6], YOU SAID THAT [3], SDA [41], DAVS [5], ATVGNET [2] AND YI'S METHOD [9]). THE NUMBER IN BRACKET IS THE RANKING OF METHOD.

Metric	Wav2Lip [26]	X2Face [6]	YST [3]	SDA [41]	DAVS [5]	ATVGnet [2]	Yi's Method [9]	Ours
PSNR ↑	30.72	29.82	29.91	29.44	29.81	30.91	30.85	**31.98** (1)
SSIM ↑	0.76	0.75	0.77	0.68	0.73	**0.81**	0.75	0.81 (1)
LMD ↓	1.61	1.60	1.63	2.32	1.73	**1.37**	1.58	1.44 (2)

neither head motion nor eye blink.

SDA. SDA [41] generates talking-face video from input video and audio using a temporal GAN with 2 discriminators, i.e. frame discriminator and sequence discriminator, which are designed for different aspects of a video. However, the quality of generation results decreases over time. It can not keep the boundary of the cropped face so results of SDA are talking-face videos without background. The resolution of its generation results is 96×128, which is lower than ours.

Rendering-based Methods. In recent years, several rendering-based methods (e.g. [9]–[11], [39]) for generating talking-face video from a pair of unmatched video and audio have been proposed. Their pipelines are similar, which use the 3D parametric model as a prior, whose parameters are composed of identity components and expression components. They recover identity parameters from the video input by face reconstruction methods and predict expression parameters from the audio input by neural networks. Then they render the reconstructed face model and obtain a rendered frame. Most methods are trained for only one person, and need a large number of training data of one specified person. Only Yi's method [9] trained a general model and works for arbitrary identity. The model is trained on a dataset with many identities and fine-tuned on a short video of the person. It also works without fine-tuning. Therefore, we only compare our method with Yi's method. In quantitative comparison, Yi's Method is trained on LRW dataset without fine-tuning.

Qualitative Comparison. The qualitative comparison between the STOA methods (Wav2Lip [26], X2Face [6], You said that [3], SDA [41], DAVS [5], ATVGnet [2] and Yi's Method [9]) and ours is shown in Figure 10. Results of SDA are apparently worse than others. X2Face distorts the shape of face and changes the identity. The results of ATVGnet, DAVS and You Said That have no head motion and no eye blink. ATVGnet results are of low resolution and degraded visual quality. Wav2Lip results are of low definition in facial area, which sometimes causes an obvious boundary between facial part and other parts when directly covering the generated region onto the background. Yi's Method can not generate talking-head videos in real time. While our method can generate high-quality talking-head videos with head motion in real time.

Quantitative Comparison. We use the test set of LRW dataet [48] for quantitative metric evaluation. For each video in the test set, we take its first frame and audio signal as inputs, and generate a talking-face video for each comparison method. We compare the results with the ground-truth videos, after aligning them according to the way used in ATVGnet. We use Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Metrics (SSIM) to evaluate the quality of images, and use Landmark Distance (LMD) to evaluate the accuracy of lip movement. The results of quantitative comparison between ours and the SOTA methods (Wav2Lip [26], X2Face [6], You said that [3], SDA [41], DAVS [5], ATVGnet [2] and Yi's Method [9]) are summarized in Table II, showing that our method has better performance than most of the SOTA methods on the three metrics above.

Perceptual Study. There is no universal metric to evaluate the visual quality of generated video. The metrics used above are also limited in predicting visual quality. Therefore, it is a good way to use a perceptual study for measuring the visual quality. We collected 10 videos in the wild with different head poses and 10 audios as inputs, and combined them to generate 100 videos. We used ATVGnet, Wav2Lip, X2Face, YST and our method to generate talking-face videos without background. A group of five videos generated from the same input was presented in a random order to the participants and they were asked to select the video with the best visual quality (VQ), lip synchronization (LS), inter-frame continuity (IFC) and overall quality (Overall): (1) visual quality is to measure the definition and naturalness of videos, (2) lip synchronization is to measure the correspondence between lip movements and audios, (3) inter-frame continuity is to measure continuity between successive frames of videos, and (4) overall quality is to measure videos by combining all the three metrics. 20 participants attended the perceptual study and each of them compared 20 random groups of video. The statistics of the user study are summarized in Table III. Our method has the best visual quality, lip synchronization, inter-frame continuity and overall quality.

TABLE III
PERCEPTUAL STUDY ON VISUAL QUALITY, LIP SYNCHRONIZATION, AND INTER-FRAME CONTINUITY.

Methods	VQ	LS	IFC	Overall
ATVGnet [2]	1.0%	1.5%	1.0%	1.0%
Wav2Lip [26]	18.0%	41.2%	29.5%	24.5%
X2Face [6]	0.0%	0.2%	0.0%	0.0%
YST [3]	0.8%	1.2%	2.0%	1.5%
Ours	**80.2%**	**55.8%**	**67.5%**	**73.0%**

In addition to the figures illustrated in this section, video examples are presented in the accompanying demo video.

F. Running time

Our method can generate talking-face videos in real time. For generating a $6s$ video with 159 frames, it takes $0.75s$ in generating facial video with 256×256 resolution and it takes $2.62s$ in total in generating a video with 1280×720

TABLE IV
RUNNING TIME OF OURS AND THE STOA METHODS (WAV2LIP [26], X2FACE [6], YOU SAID THAT [3], SDA [41], DAVS [5], ATVGNET [2] AND YI'S METHOD [9]). YOU SAID THAT, ATVGNET AND X2FACE CHANGE THE HEAD POSE SO THEY CANNOT GENERATE RESULTS WITH BACKGROUND. SDA AND DAVS CANNOT KEEP THE BOUNDARY OF THE CROPPED FACE SO THEY ALSO CANNOT GENERATE RESULTS WITH BACKGROUND.

Metric	Wav2Lip [26]	X2Face [6]	YST [3]	SDA [41]	DAVS [5]	ATVGnet [2]	Yi's Method [9]	Ours
Time Cost per Frame (w/o BG) (ms)	9.9	95.0	88.9	32.5	107.2	6.7	567.4	**4.7**
Maximum FPS (w/o BG)	100	11	11	31	9	16	2	**212**
Maximum FPS (with BG)	42	/	/	/	/	/	1	**60**

background. For generating a video with the same input, (1) ATVGnet [2] takes $1.07s$ to generate video without background, (2) You Said That [3] takes $14.13s$ to generate video without background, (3) X2Face [6] takes $15.10s$ to generate video without background, (4) DAVS [5] takes $17.05s$ to generate video without background, and (5) SDA [41] takes $5.17s$ to generate video without background. Yi's method [9] takes $90.22s$ to generate video without background and $127.24s$ to generate video with background, which also takes about 1 hour to fine-tune their network. Wav2Lip [26] takes $1.58s$ to generate video without background and takes $3.83s$ to direct cover it with background. However, sometimes the definition of facial area is lower than other parts which causes an obvious boundary between foreground and background. Therefore, only our method and Wav2Lip can generate talking-face video with background in real time (over 25 fps). We show running time of all above methods in TABLE IV for a direct comparison.

VII. CONCLUSION

We propose a novel fully convolutional network with DCKs for the multi-modal task of audio-driven taking face video generation. Our simple yet effective system can generate high-quality talking-face video from unmatched video and audio in real time. Our solution is end-to-end, one-for-all and robust to different identities, head postures and audios. For preserving identities in both input and output talking-head videos, we propose a novel supervised training scheme. The results show our method can generate high-quality 60 fps talking-head video with background in real time. Comparison and evaluation between our method and state-of-the-art methods show that our method achieves a good balance between various criteria such as running time, qualitative and quantitative qualities. Our novel DCK technique can potentially be applied to other multi-modal generation tasks, and meanwhile, our theoretical interpretation of DCK can be extended from fully convolutional network to forward networks involving ResNet modules, which we will investigate in future work.

ACKNOWLEDGMENT

This work was partially supported by the Natural Science Foundation of China (61725204).

REFERENCES

[1] O. Fried, A. Tewari, M. Zollhöfer, A. Finkelstein, E. Shechtman, D. B. Goldman, K. Genova, Z. Jin, C. Theobalt, and M. Agrawala, "Text-based editing of talking-head video," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 68:1–68:14, 2019.

[2] L. Chen, R. K. Maddox, Z. Duan, and C. Xu, "Hierarchical cross-modal talking face generation with dynamic pixel-wise loss," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7832–7841.

[3] J. S. Chung, A. Jamaludin, and A. Zisserman, "You said that?" in *British Machine Vision Conference (BMVC)*, 2017.

[4] Y. Song, J. Zhu, D. Li, A. Wang, and H. Qi, "Talking face generation by conditional recurrent adversarial network," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019, pp. 919–925.

[5] H. Zhou, Y. Liu, Z. Liu, P. Luo, and X. Wang, "Talking face generation by adversarially disentangled audio-visual representation," in *The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*, 2019, pp. 9299–9306.

[6] O. Wiles, A. S. Koepke, and A. Zisserman, "X2face: A network for controlling face generation using images, audio, and pose codes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 690–706.

[7] B. Fan, L. Wang, F. K. Soong, and L. Xie, "Photo-real talking head with deep bidirectional LSTM," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4884–4888.

[8] K. Vougioukas, S. Petridis, and M. Pantic, "Realistic speech-driven facial animation with gans," *International Journal of Computer Vision*, pp. 1–16, 2019.

[9] R. Yi, Z. Ye, J. Zhang, H. Bao, and Y.-J. Liu, "Audio-driven talking face video generation with natural head pose," *arXiv preprint arXiv:2002.10137*, 2020.

[10] L. Song, W. Wu, C. Qian, R. He, and C. C. Loy, "Everybody's talkin': Let me talk as you want," *arXiv preprint arXiv:2001.05201*, 2020.

[11] J. Thies, M. Elgharib, A. Tewari, C. Theobalt, and M. Nießner, "Neural voice puppetry: Audio-driven facial reenactment," in *European Conference on Computer Vision*. Springer, 2020, pp. 716–731.

[12] H. Kim, P. Garrido, A. Tewari, W. Xu, J. Thies, M. Nießner, P. Pérez, C. Richardt, M. Zollhöfer, and C. Theobalt, "Deep video portraits," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 163:1–163:14, 2018.

[13] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of RGB videos," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2387–2395.

[14] H. Averbuch-Elor, D. Cohen-Or, J. Kopf, and M. F. Cohen, "Bringing portraits to life," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 6, pp. 1–13, 2017.

[15] A. Pumarola, A. Agudo, A. M. Martínez, A. Sanfeliu, and F. Moreno-Noguer, "GANimation: Anatomically-aware facial animation from a single image," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 835–851.

[16] E. Zakharov, A. Shysheya, E. Burkov, and V. S. Lempitsky, "Few-shot adversarial learning of realistic neural talking head models," *CoRR*, vol. abs/1905.08233, 2019.

[17] Y. Zhang, S. Zhang, Y. He, C. Li, C. C. Loy, and Z. Liu, "One-shot face reenactment," *CoRR*, vol. abs/1908.03251, 2019.

[18] J. R. Nazzaro and J. N. Nazzaro, "Auditory versus visual learning of temporal patterns," *Journal of Experimental Psychology*, vol. 84, no. 3, pp. 477–8, 1970.

[19] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1999, pp. 187–194.

[20] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "Facewarehouse: A 3D facial expression database for visual computing," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 3, pp. 413–425, 2014.

[21] D. Ha, A. M. Dai, and Q. V. Le, "Hypernetworks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[22] J. Z. Esquivel, A. C. Vargas, P. L. Meyer, and O. Tickoo, "Adaptive convolutional kernels," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 2019, pp. 1998–2005.
[23] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8971–8980.
[24] X. Nie, Y. Li, L. Luo, N. Zhang, and J. Feng, "Dynamic kernel distillation for efficient pose estimation in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 6942–6950.
[25] Y. Chen, X. Dai, M. Liu, D. Chen, L. Yuan, and Z. Liu, "Dynamic convolution: Attention over convolution kernels," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 030–11 039.
[26] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. V. Jawahar, "A lip sync expert is all you need for speech to lip generation in the wild," in *The 28th ACM International Conference on Multimedia (MM)*, 2020, pp. 484–492.
[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015, pp. 234–241.
[28] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.
[29] D. L. Hall and J. Llinas, "An introduction to multisensor data fusion," *Proceedings of the IEEE*, vol. 85, no. 1, pp. 6–23, 1997.
[30] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli, "Multimodal fusion for multimedia analysis: a survey," *Multimedia systems*, vol. 16, no. 6, pp. 345–379, 2010.
[31] T. Baltrusaitis, C. Ahuja, and L. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 423–443, 2019.
[32] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi, "Attention-based multimodal fusion for video description," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 4193–4202.
[33] B. Nojavanasghari, D. Gopinath, J. Koushik, T. Baltrušaitis, and L.-P. Morency, "Deep multimodal fusion for persuasiveness prediction," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, 2016, pp. 284–288.
[34] S. Lin, M. Bai, F. Liu, L. Shen, and Y. Zhou, "Orthogonalization-guided feature fusion network for multimodal 2d+ 3d facial expression recognition," *IEEE Transactions on Multimedia*, vol. 23, pp. 1581–1591, 2021.
[35] S. Wang, L. Hao, and Q. Ji, "Knowledge-augmented multimodal deep regression bayesian networks for emotion video tagging," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 1084–1097, 2020.
[36] Q. Kuang, X. Jin, Q. Zhao, and B. Zhou, "Deep multimodality learning for uav video aesthetic quality assessment," *IEEE Transactions on Multimedia*, vol. 22, no. 10, pp. 2623–2634, 2020.
[37] P. Buitelaar, I. D. Wood, S. Negi, M. Arcan, J. P. McCrae, A. Abele, C. Robin, V. Andryushechkin, H. Ziad, H. Sagha *et al.*, "Mixedemotions: An open-source toolbox for multimodal emotion analysis," *IEEE Transactions on Multimedia*, vol. 20, no. 9, pp. 2454–2465, 2018.
[38] Y.-G. Jiang, Z. Wu, J. Tang, Z. Li, X. Xue, and S.-F. Chang, "Modeling multimodal clues in a hybrid deep learning framework for video classification," *IEEE Transactions on Multimedia*, vol. 20, no. 11, pp. 3137–3147, 2018.
[39] X. Wen, M. Wang, C. Richardt, Z.-Y. Chen, and S.-M. Hu, "Photorealistic audio-driven video portraits," *IEEE Transactions on Visualization and Computer Graphics*, 2020.
[40] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, "Least squares generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2794–2802.
[41] K. Vougioukas, S. Petridis, and M. Pantic, "End-to-end speech-driven facial animation with temporal GANs," in *British Machine Vision Conference (BMVC)*, 2018, p. 133.
[42] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision (ICCV)*, 2017, pp. 2223–2232.
[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
[44] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.
[45] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS 2017 Autodiff Workshop: The Future of Gradient-based Machine Learning Software and Techniques*, 2017.
[46] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. Jul, pp. 1755–1758, 2009.
[47] X. Zhang, X. Wu, X. Zhai, X. Ben, and C. Tu, "Davd-net: Deep audio-aided video decompression of talking heads," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12 335–12 344.
[48] J. S. Chung and A. Zisserman, "Lip reading in the wild," in *13th Asian Conference on Computer Vision (ACCV)*, 2016, pp. 87–103.
[49] L. Chen, Z. Li, R. K. Maddox, Z. Duan, and C. Xu, "Lip movements generation at a glance," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 538–553.

Zipeng Ye is a Ph.D student with Department of Computer Science and Technology, Tsinghua University. He received his B.Eng. degree from Tsinghua University, China, in 2017. His research interests include multi-modal perception and deep learning in virtual human modeling.

Mengfei Xia is a PhD student with Department of Computer Science and Technology, Tsinghua University. He won the silver medal twice in 30th and 31st National Mathematical Olympiad of China. His research interests include mathematical foundation in deep learning, image processing and computer vision.

Ran Yi is an Assistant Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. She received the BEng degree and the PhD degree from Tsinghua University, China, in 2016 and 2021. Her research interests include computer vision, computer graphics and computational geometry.

Juyong Zhang is an Associate Professor in the School of Mathematical Sciences at University of Science and Technology of China. He received the BS degree from the University of Science and Technology of China in 2006, and the PhD degree from Nanyang Technological University, Singapore. His research interests include computer graphics, computer vision, and numerical optimization. He is an associate editor of IEEE Trans. on Multimedia.

Yu-Kun Lai is a Professor at School of Computer Science and Informatics, Cardiff University, UK. He received his B.S and PhD degrees in Computer Science from Tsinghua University, in 2003 and 2008 respectively. His research interests include computer vision, geometric modeling and image processing. For more information, visit https://users.cs.cf.ac.uk/Yukun.Lai/

Xuwei Huang received the bachelor degree in Information Engineer and the master degree in Electronic Engineer, in 2012 and 2015, respectively, both from South China University of Technology, GuangZhou, China. Currently, he is an AI algorithm engineer in Beijing Kuaishou Technology Co, working on computer vision and speech synthesis.

Guoxin Zhang is currently with Kuaishou Technology. He received the B.E degree and the Ph.D degree from Tsinghua University, China, in 2007 and 2012. His research interests include multimedia, computer graphics and computer vision.

Yong-Jin Liu is a Professor with the Department of Computer Science and Technology, Tsinghua University, China. He received the B.Eng degree from Tianjin University, China, in 1998, and the PhD degree from the Hong Kong University of Science and Technology, Hong Kong, China, in 2004. His research interests include multimedia, computer graphics and computer vision. For more information, visit https://cg.cs.tsinghua.edu.cn/people/~Yongjin/Yongjin.htm