

CodeTalker: Speech-Driven 3D Facial Animation with Discrete Motion Prior

Jinbo Xing^{*1} Menghan Xia² Yuechen Zhang¹ Xiaodong Cun² Jue Wang² Tien-Tsin Wong¹

¹The Chinese University of Hong Kong ²Tencent AI Lab

Abstract

Speech-driven 3D facial animation has been widely studied, yet there is still a gap to achieving realism and vividness due to the highly ill-posed nature and scarcity of audio-visual data. Existing works typically formulate the cross-modal mapping into a regression task, which suffers from the regression-to-mean problem leading to over-smoothed facial motions. In this paper, we propose to cast speech-driven facial animation as a code query task in a finite proxy space of the learned codebook, which effectively promotes the vividness of the generated motions by reducing the cross-modal mapping uncertainty. The codebook is learned by self-reconstruction over real facial motions and thus embedded with realistic facial motion priors. Over the discrete motion space, a temporal autoregressive model is employed to sequentially synthesize facial motions from the input speech signal, which guarantees lip-sync as well as plausible facial expressions. We demonstrate that our approach outperforms current state-of-the-art methods both qualitatively and quantitatively. Also, a user study further justifies our superiority in perceptual quality. The source code will be publicly available.

1. Introduction

3D facial animation has been an active research topic for decades, as attributed to its broad applications in virtual reality, film production, and games. The high correlation between speech and facial gestures (especially lip movements) makes it possible to drive the facial animation with a speech signal. Early attempts are mainly made to build the complex mapping rules between phonemes and their visual counterpart, which usually have limited performance [54, 66]. With the advances in deep learning, recent speech-driven facial animation techniques push forward the state-of-the-art significantly. However, it still remains challenging to generate human-like motions.

As an ill-posed problem, speech-driven facial animation generally has multiple plausible outputs for every input.

Such ambiguity tends to cause over-smoothed results. Anyhow, person-specific approaches [31, 50] can usually obtain decent facial motions because of the relatively consistent talking style, but have low scalability to general applications. Recently, VOCA [10] extends these methods to generalize across different identities, however, they generally exhibit mild or static upper face expressions. This is because VOCA formulates the speech-to-motion mapping as a regression task, which encourages averaged motions, especially in the upper face that is only weakly or even uncorrelated to the speech signal. To reduce the uncertainty, FaceFormer [16] utilizes long-term audio context through a transformer-based model and synthesizes the sequential motions in an autoregressive manner. Although it gains important performance promotion, it still inherits the weakness of one-to-one mapping formulation and suffers from a lack of subtle high-frequency motions. Differently, MeshTalk [51] models a categorical latent space for facial animation that disentangles audio-correlated and audio-uncorrelated information so that both aspects could be well handled. Anyway, the categorical latent space is represented by some proxy numbers instead of an explicit codebook with feature vectors, which makes the training tricky and hence impedes its performance.

We get inspiration from 3D Face Morphable Model (3DMM) [37], where general facial expressions are represented in a low-dimensional space. Accordingly, we propose to formulate speech-driven facial animation as a code query task in a finite proxy space of the learned discrete codebook prior. The codebook is learned by self-reconstruction over real facial motions using a vector-quantized autoencoder (VQ-VAE) [60], which along with the decoder stores the realistic facial motion priors. In contrast to the continuous linear space of 3DMM, combinations of codebook items form a discrete prior space with only finite cardinality. Still, in the context of the decoder, the code representation possesses high expressiveness. Through mapping the speech to the finite proxy space, the uncertainty of the speech-to-motion mapping is significantly attenuated and hence promotes the quality of motion synthesis. Conceptually, the proxy space approximates the facial motion space, where the learned codebook items

^{*} Work done during an internship at Tencent AI Lab.

serve as discrete motion primitives.

Based on the learned discrete codebook, we propose a code-query-based temporal autoregressive model for speech-conditioned facial motion synthesis, called *CodeTalker*. Specifically, taking a speech signal as input, our model predicts the motion feature tokens in a temporal causal manner. Then, the feature tokens are used to query the code sequence in the discrete space, followed by facial motion reconstruction. Thanks to the contextual modeling over history motions and cross-modal alignment, the proposed CodeTalker shows the advantages of achieving accurate lip motions and natural expressions. Extensive experiments show that the proposed CodeTalker demonstrates superior performance on existing datasets. Systematic studies and experiments are conducted to demonstrate the merits of our method over previous works. The contributions of our work are as follows:

- We make the first attempt to model the facial motion space with discrete primitives, which offers advantages to promote motion synthesis realism against cross-modal uncertainty.
- We propose a discrete motion prior based temporal autoregressive model for speech-driven facial animation, which outperforms existing state-of-the-art methods.
- We will make our code fully available, including the data processing, metrics, trained models, *etc.*, to standardize the speech-driven 3D facial animation task for follow-ups.

2. Related Works

2.1. Speech-driven 3D Facial Animation

Computer facial animation is a long-standing task [45] and has attracted rapidly increased interest over the past decades [5, 20, 32, 34, 36, 55, 65, 72]. As a branch, speech-driven facial animation is to reenact a person in sync with input speech sequences. While extensive literature in this field works on 2D talking heads [1, 7–9, 11, 23, 28, 29, 38, 39, 48, 52, 62, 64, 68, 69], we focus on facial animation on 3D models in this work, which can be roughly categorized into linguistics-based and learning-based methods.

Linguistics-based methods. Typically, linguistics-based methods [12, 42, 54, 66] establish a set of complex mapping rules between phonemes and their visual counterparts, *i.e.*, visemes [19, 35, 43]. For example, the dominance function [42] is to determine the influence of phonemes on the respective facial animation control parameters. Xu *et al.* [66] defines animation curves for a constructed canonical set of visemes to generate synchronized mouth movements. There are also some methods considering the many-to-many mapping between phonemes and visemes, as demon-

strated in the dynamic visemes model [54] and, more recently, the JALI [12]. Based on psycholinguistic considerations and built upon the Facial Action Coding System (FACS) [13], JALI factors mouth movements into lip and jaw rig animation and generate compelling co-articulation results. Although these methods have explicit control over the animation, they have complex procedures and lack a principled way to animate the entire face.

Learning-based methods. Learning-based methods [6, 10, 16, 17, 24, 31, 40, 47, 53, 63] resort to a data-driven framework. Cao *et al.* [6] achieve emotional lip sync by the proposed constrained search and Anime Graph structure. Recently, Taylor *et al.* [53] propose a deep-learning-based model utilizing a sliding window approach on the transcribed phoneme sequences input. Karras *et al.* [31] propose a convolution-based network with a learnable emotion database to animate a speech-driven 3D mesh. More recently, VisemeNet [71] employ a three-stage Long Short-Term Memory (LSTM) network to predict the animation curve for a lower face lip model.

We review the most related works more concretely here as they have the same setting as this work, *i.e.*, training on high-resolution paired audio-mesh data and speaker-independently animating entire face meshes in vertex space. MeshTalk [51] successfully disentangles audio-correlated and uncorrelated facial information with a categorical latent space. However, the latent space adopted is not optimal with limited expressiveness, thus the animation quality is not stable when applied in a data-scarcity setting. VOCA [10] employs powerful audio feature extraction models and can generate facial animation with different speaking styles. Furthermore, FaceFormer [16] considers long-term audio context with transformer [61] rendering temporally stable animations. Despite the appealing animations, both suffer from the over-smoothing problem, as they directly regress the facial motion in the highly ill-posed audio-visual mapping with large uncertainty and ambiguity.

2.2. Discrete Prior Learning

In the last decades, discrete prior representation with learned dictionaries has demonstrated its superiority in image restoration tasks [14, 22, 30, 56, 57], since clear image details are well-preserved in the dictionaries. This line of techniques further inspires the high-capacity and high-compressed discrete prior learning. VQ-VAE [60] first presents to learn discrete representations (codebook) of images and autoregressively model their distribution for image synthesis. The follow-up works, VQ-VAE2 [49] and VQ-GAN [15] further improve the quality of high-resolution image synthesis. Recently, discrete prior learning and its variants have been exploited for image colorization [26], inpainting [46], blind face restoration [70], text-to-image synthesis [21], *etc.*

In addition to the image modality, most recent works also explore the power of discrete prior learning in tasks with other modalities, such as dyadic face motion generation [44], co-speech gesture synthesis [2], speech enhancement [67]. Inspired by codebook learning, this work investigates to learn discrete motion prior for speech-driven 3D facial animation. Different from [44], we exploit the discrete motion primitives for facial motion representation in a context-rich manner, which is more effective to learn general priors.

3. Method

We aim to synthesize sequential 3D facial motions from a speech signal, so that any neutral face mesh could be animated as a lip-synchronized talking face. However, this is an ill-posed problem since one speech could be matched by multiple potential facial animations. Such ambiguity tends to make cross-modal learning suffer from averaged motions and lack of subtle variations. To bypass this barrier, we propose to first model the facial motion space with the learned discrete motion prior, and then learn a speech-conditioned temporal autoregressive model over this space, which promotes robustness against the cross-modal uncertainty.

Formulation. Let $\mathbf{M}_{1:T} = (\mathbf{m}_1, \dots, \mathbf{m}_T)$ be a sequence of facial motions, where each frame $\mathbf{m}_t \in \mathbb{R}^{V \times 3}$ denotes the 3D movement of V vertices over a neutral-face mesh template $\mathbf{h} \in \mathbb{R}^{V \times 3}$. Let further $\mathbf{A}_{1:T} = (\mathbf{a}_1, \dots, \mathbf{a}_T)$ be a sequence of speech snippets, each of which $\mathbf{a}_t \in \mathbb{R}^d$ has d samples to align with the corresponding (visual) frame \mathbf{m}_t . Then, our goal is to sequentially synthesize $\mathbf{M}_{1:T}$ from $\mathbf{A}_{1:T}$ so that an arbitrary neutral facial template \mathbf{f} could be animated as $\mathbf{H}_{1:T} = \{\mathbf{m}_1 + \mathbf{h}, \dots, \mathbf{m}_T + \mathbf{h}\}$.

3.1. Discrete Facial Motion Space

Visual realistic facial animations should present accurate lip motions and natural expressions. To achieve this from speech signals, extra motion priors are required to reduce the uncertainty and complement realistic motion components. As witnessed by the recent image restoration task [70], discrete codebook prior [60] demonstrates advantages in guaranteeing high-fidelity results even from a severely degraded input. Inspired by this, we propose to model the facial motion space as a discrete codebook by learning from tracked real-world facial motions.

Codebook of motion primitives. We manage to learn a codebook $\mathcal{Z} = \{\mathbf{z}_k \in \mathbb{R}^C\}_{k=1}^N$ that allows any facial motion \mathbf{m}_t to be represented by a group of allocated items $\{\mathbf{z}_k\}_{k \in \mathcal{S}}$, where \mathcal{S} denotes the selected index set through Eq. 1. Conceptually, the codebook items serve as the motion primitives of a facial motion space. To this end, we pre-train a transformer-based VQ-VAE that consists of an encoder E , a decoder D , and a context-rich codebook \mathcal{Z} , under the

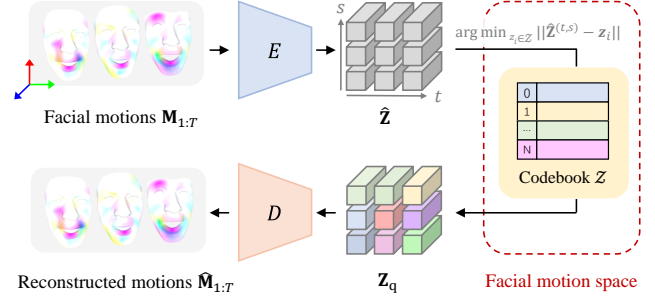


Figure 1. Learning framework of facial motion space. The learned motion primitives, as embedded in the codebook, serve to represent the facial motions in a spatial and temporal manner.

self-reconstruction of realistic facial motions. As shown in Figure 1, the facial motions $\mathbf{M}_{1:T}$ is first embedded as a temporal feature $\hat{\mathbf{Z}} = E(\mathbf{M}_{1:T}) \in \mathbb{R}^{T' \times H \times C}$, where H is the number of face components and T' denotes the number of encoded temporal units ($P = \frac{T}{T'}$ frames). Then, we obtain the quantized motion sequence $\mathbf{Z}_q \in \mathbb{R}^{T' \times H \times C}$ via an element-wise quantization function $Q(\cdot)$ that maps each item in $\hat{\mathbf{Z}}$ to its nearest entry in codebook \mathcal{Z} :

$$\mathbf{Z}_q = Q(\hat{\mathbf{Z}}) := \arg \min_{\mathbf{z}_k \in \mathcal{Z}} \|\hat{\mathbf{z}}_t - \mathbf{z}_k\|_2. \quad (1)$$

Then, the self-reconstruction is given by:

$$\hat{\mathbf{M}}_{1:T} = D(\mathbf{Z}_q) = D(Q(E(\mathbf{M}_{1:T}))). \quad (2)$$

Note that, the discrete facial motion space reduces the mapping ambiguity with the finite cardinality, but never sacrifices its expressiveness thanks to its context-rich representation as a latent space.

Training objectives. To supervise the quantized autoencoder training, we adopt a motion-level loss and two intermediate code-level losses:

$$\mathcal{L}_{VQ} = \|\mathbf{M}_{1:T} - \hat{\mathbf{M}}_{1:T}\|_1 + \|\text{sg}(\hat{\mathbf{Z}}) - \mathbf{Z}_q\|_2^2 + \beta \|\hat{\mathbf{Z}} - \text{sg}(\mathbf{Z}_q)\|_2^2, \quad (3)$$

where the first term is a reconstruction loss, the latter two are adopted to update the codebook items by reducing the distance between the codebook \mathcal{Z} and embedded features $\hat{\mathbf{Z}}$. $\text{sg}(\cdot)$ stands for a stop-gradient operation and β is a weighting factor controlling the update rate of the codebook and encoder. Since the quantization function (Eq. 1) is not differentiable, the straight-through gradient estimator [4, 60] is employed to copy the gradients from the decoder input to the encoder output.

Discussion. Recently, Learn2Listen [44] has applied VQ-VAE for facial expression synthesis in response to a given talking head harnessing 2D monocular videos to obtain 3DMM coefficients. In addition to distinct applications,

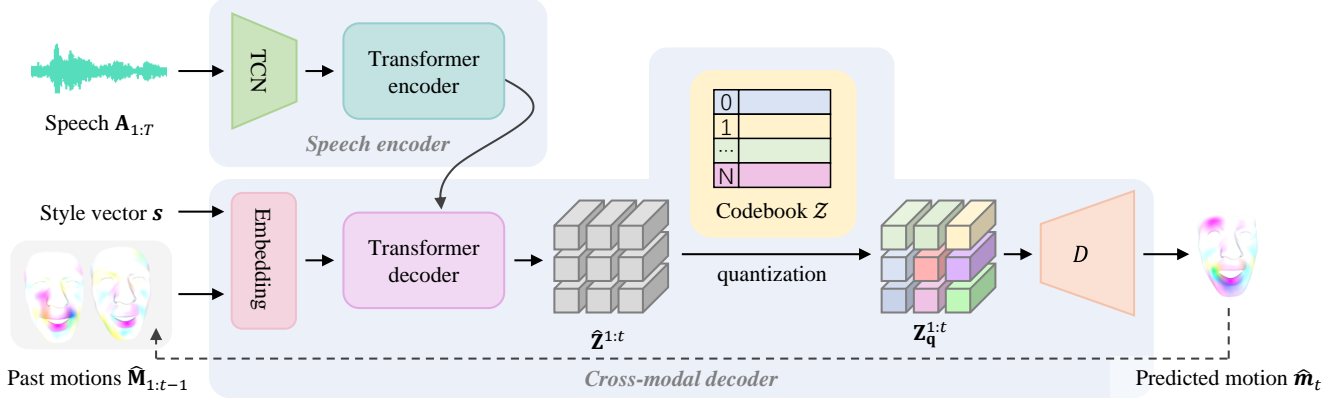


Figure 2. Diagram of our speech-driven motion synthesis model. Given the speech $A_{1:T}$ and style vector s as input, the model learn to recursively generate a sequence of facial motions by predicting the motion codes. As embedded with well-learned motion priors, the pre-trained codebook and decoder are frozen during training.



Figure 3. Concept comparison with Learn2Listen [44]. (Top) The speaker-specific facial expression coefficient prior [44], in which each code represents a sequence of facial expression coefficients. (Bottom) Our speaker-agnostic generic motion prior, in which each code represents the motion primitive of face components. The blue dotted boxes indicate what information each code may represent conceptually.

here we would like to emphasize our major differences. First, Learn2Listen constructs speaker-specific codebooks while ours uses a generic codebook that is feasible to represent arbitrary facial motions. Since cross-character motions are absorbed, our codebook is naturally embedded with more plentiful priors. Second, Learn2Listen utilizes the codebook to represent common sequences of facial expressions by the way of 3DMM coefficients, *i.e.*, each code represents a sequence (8 frames) of facial expressions. Differently, our codebook is formulated to represent the vertex-based facial motion space, where the codes are embedded with per-vertex motions of facial components and represent the facial motion (within a temporal unit) in a context-rich manner. As compared in Figure 3, the codebook of Learn2Listen is learned to memorize typical sequential facial expressions of a specific speaker within 3DMM space, which cannot synthesize realistic facial motions with subtle details due to the limited expressiveness of 3DMM and is bounded by the accuracy of 3D reconstruction techniques [44]. As the first attempt, our codebook is learned to represent the generic facial motion space with motion prim-

itives for captured facial mesh data, which is more effective to embed general priors preserving vivid facial details.

We further discuss the hyper-parameters of the codebook. First, the length of the temporal unit P and the number of face components H determine the complexity of the motion primitives in temporal and spatial aspects respectively. Generally, complex motion primitives cause low flexibility and reusability and thus hinder representation effectiveness. On the opposite, overly simple motion primitives challenge motion prediction due to the lack of semantics. Besides, the codebook size N and the feature dimension C determine the representation capability, which should be defined according to the complexity of the dataset and in cooperation with P and H . In our experiment, we set $N = 256$, $P = 1$, $H = 8$ or $H = 16$, and $C = 64$ or $C = 128$ depending on the dataset, which lead to high-quality results as justified by the ablation studies in Section 4.5. More details can be found in the *Supplement*.

3.2. Speech-Driven Motion Synthesis

With the learned discrete motion prior, we can build a cross-modal mapping from the input speech to the target motion codes that could be further decoded into realistic facial motions. Along with the speech, we further adopt a control on the talking styles as input, *i.e.*, a style vector $s \in \mathbb{R}_+^M \cup \{0\}$, where M is the dimension of the learned style space (see Eq. 4). Conditioning on the speech $A_{1:T}$ and the style vector s , a temporal autoregressive model, composed of a speech encoder E_{speech} and a cross-modal decoder $D_{\text{cross-modal}}$, is employed to learn over the facial motion space, as depicted in Figure 2.

Following FaceFormer [16], our speech encoder adopts the architecture of the state-of-the-art self-supervised pre-trained speech model, wav2vec 2.0 [3], which consists of an audio feature extractor and a multi-layer transformer encoder. The audio feature extractor converts the speech of raw waveform into feature vectors through a temporal con-

volution network (TCN). Benefiting from the effective attention scheme, the transformer encoder converts the audio features into contextualized speech representations. Apart from the pre-trained codebook and VQ-VAE decoder, our cross-modal decoder contains an embedding block and a multi-layer transformer decoder with causal self-attention. The embedding block combines the past facial motions and the style embedding via:

$$\mathbf{F}_{\text{emb}}^{1:t-1} = \mathcal{P}_\theta(\hat{\mathbf{M}}_{1:t-1}) + \mathbf{B} \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|_1}, \quad (4)$$

where \mathcal{P}_θ is a linear projection layer, and $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_M] \in \mathbb{R}^{C \times M}$ denotes the M learnable basis vectors that span the style space linearly. Alike to FaceFormer [16], we equip the transformer decoder with causal self-attention to learn the dependencies between each frame in the context of the past facial motion sequence, and with cross-modal attention to align the audio and motion modalities. The output features $\hat{\mathbf{Z}}^{1:t}$ is further quantized into $\mathbf{Z}_q^{1:t}$ via Eq. 1 and decoded by the pre-trained VQ-VAE decoder. The newly predicted motion $\hat{\mathbf{m}}_t$ is used to update the past motions as $\hat{\mathbf{M}}_{1:t}$, in preparation for the next prediction. Formally, this recursive process can be written as:

$$\hat{\mathbf{m}}_t = D_{\text{cross-modal}}(E_{\text{speech}}(\mathbf{A}_{1:T}), \mathbf{s}, \hat{\mathbf{M}}_{1:t-1}). \quad (5)$$

Training objectives. We train the transformer encoder, decoder and the embedding block for cross-modality mapping, while keeping the codebook \mathcal{Z} and motion decoder D frozen. To benefit from the speech representation learning from large-scale corpora, we initialize the TCN and transformer encoder with the pre-trained wav2vec 2.0 weights. Overall, the autoregressive model is trained in a teaching-forcing scheme, under the constraint of two loss terms: (i) feature regularity loss \mathcal{L}_{reg} measuring the deviation between the predicted motion feature $\hat{\mathbf{Z}}^{1:T}$ and the quantized feature $\mathbf{Z}_q^{1:T}$ from codebook, and (ii) motion loss $\mathcal{L}_{\text{motion}}$ measuring the difference between the predicted motions $\hat{\mathbf{M}}_{1:T}$ and the ground-truth motions $\mathbf{M}_{1:T}$, which plays an important role to stabilize the training process. The final loss function is:

$$\begin{aligned} \mathcal{L}_{\text{syn}} &= \mathcal{L}_{\text{reg}} + \mathcal{L}_{\text{motion}} \\ &= \|\hat{\mathbf{Z}}^{1:T} - \text{sg}(\mathbf{Z}_q^{1:T})\|_2^2 + \|\hat{\mathbf{M}}_{1:T} - \mathbf{M}_{1:T}\|_2^2. \end{aligned} \quad (6)$$

3.3. Training Details

At stage one, we train the VQ-VAE model (Figure 1) on a single NVIDIA V100 for 200 epochs (~ 2 hours) with the AdamW [41] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-8$), where the learning rate is initialized as 10^{-4} , and the mini-batch size is set as 1. At stage two, we train the temporal autoregressive model with the Adam optimizer [33]. The training duration is 150 epochs (~ 5 hours) and other hyper-parameters remain unchanged as stage one.

Style embedding space. The style embedding space is linearly spanned by M learned basis vectors, where each style is represented by a style vector \mathbf{s} that serves as the linear combination coefficients or a coordinate. During training, we assign each speaker (e.g. no. i) with a standard unit vector \mathbf{e}_i as a style vector, under the assumption that each speaker is associated with a unique and consistent style. Anyway, arbitrary style vectors are allowed to interpolate new talking styles during inference.

4. Experiments

4.1. Datasets and Implementations

We employ two widely used datasets, BIWI [18] and VOCASET [10], to train and test different methods in our experiments. Both datasets contain 4D face scans together with utterances spoken in English. BIWI contains 40 unique sentences shared across all speakers in the dataset, while VOCASET contains 255 unique sentences, which are partially shared among different speakers.

BIWI dataset. BIWI is a 3D audio-visual corpus of affective speech and facial expression in the form of dense dynamic 3D face geometries, which is originally proposed to study affective communication among humans. There is a total of 40 sentences uttered by 14 subjects, eight females and six males. Each sentence was recorded twice: with and without emotion. On average, each sentence is 4.67 seconds long. The 3D face dynamics are captured at 25fps, each with 23370 vertices and registered topology. We follow the data splits from FaceFormer [16] and only use the emotional subset. Specifically, the training set (BIWI-Train) contains 192 sentences, while the validation set (BIWI-Val) contains 24 sentences. There are two testing sets, in which BIWI-Test-A includes 24 sentences spoken by six seen subjects and BIWI-Test-B contains 32 sentences spoken by eight unseen subjects. BIWI-Test-A can be used for both quantitative and qualitative evaluation due to the seen subjects during training, while BIWI-Test-B is more suitable for qualitative evaluation.

VOCASET dataset. VOCASET is comprised of 480 paired audio-visual sequences recorded from 12 subjects. The facial motion is captured at 60fps and is about 4 seconds long. Different from BIWI, each 3D face mesh is registered to the FLAME [37] topology with 5023 vertices. We adopt the same training (VOCA-Train), validation (VOCA-Val), and testing (VOCA-Test) splits as VOCA [10] and FaceFormer for fair comparisons.

Implementations. We compare our work with three state-of-the-art methods: VOCA [10], MeshTalk [51] and FaceFormer [16]. We train and test VOCA on BIWI using the official codebase, while directly testing the released model that was trained on VOCASET. For MeshTalk, we train and test it using the official implementation on the two

Table 1. Quantitative evaluation on BIWI-Test-A. Lower means better for both metrics.

Method	Lip Vertex Error ($\times 10^{-4}$ mm)	FDD ($\times 10^{-5}$ mm)
VOCA	6.5563	8.1816
MeshTalk	5.9181	5.1025
FaceFormer	5.3077	4.6408
CodeTalker (Ours)	4.7914	4.1170

datasets. To compare with FaceFormer, we conduct testing directly using the pre-trained weights. Among the four methods, VOCA, FaceFormer and our CodeTalker require conditioning on a training speaking style during testing. For unseen subjects, we generate facial animations conditioned on all training styles. More details about the implementation details can be found in the *Supplementary Material*.

4.2. Quantitative Evaluation

Following MeshTalk [51] and FaceFormer [16], we adopt the lip vertex error to measure the lip synchronization, which is the only publicly proposed metric for speech-driven facial animation evaluation, to our best knowledge. As a complement, we introduce a new quantitative measurement, *i.e.*, upper-face motion statistics, to evaluate the overall facial dynamics.

Lip vertex error. It measures the lip deviation of a sequence with respect to the ground truth, *i.e.*, calculating the maximal L2 error of all lip vertices for each frame and takes the average over all frames.

Upper-face dynamics deviation. The upper-face expression is just loosely correlated with the speech, depending on personal talking styles and the semantics of speech content. With this belief, we propose to measure the variation of facial dynamics for a motion sequence in comparison with that of the ground truth. Specifically, the *upper-face dynamics deviation* (FDD) is calculated by:

$$\text{FDD}(\mathbf{M}_{1:T}, \hat{\mathbf{M}}_{1:T}) = \frac{\sum_{v \in \mathcal{S}_U} (\text{dyn}(\mathbf{M}_{1:T}^v) - \text{dyn}(\hat{\mathbf{M}}_{1:T}^v))}{|\mathcal{S}_U|}, \quad (7)$$

where $\mathbf{M}_{1:T}^v \in \mathbb{R}^{3 \times T}$ denotes the motions of the v -th vertex, and \mathcal{S}_U is the index set of upper-face vertices. $\text{dyn}(\cdot)$ denotes the standard deviation of the element-wise L2 norm along the temporal axis.

We calculate the lip vertex error and upper-face dynamics deviation (FDD) over all sequences in BIWI-Test-A and take the average for comparison. According to Table 1, the proposed CodeTalker achieves lower error than the existing state-of-the-arts, suggesting that it produces more accurate lip-synchronized movements. Besides, Table 1 shows that our CodeTalker achieves the best performance in terms of FDD. It indicates the high consistency between the predicted upper-face expressions together with the trend of fa-

cial dynamics (conditioned on the speech and talking styles) and those of the ground truth.

4.3. Qualitative Evaluation

We visually compare our method with other competitors in Figure 4. For fair comparison, we assign the same talking style to VOCA, FaceFormer and our CodeTalker as conditional input, which is sampled at random. To check the lip synchronization performance, we illustrate three typical frames of synthesized facial animations that speak at specific syllables, as compared in the upper partition in Figure 4. We can observe that compared with the competitors, the lip movements produced by our CodeTalker are more accurately articulated with the speech signals and also more consistent with those of the reference. For example, CodeTalker produces better lip sync with proper mouth closures when pronouncing bilabial consonant /b/ (*i.e.*, “bed-side” in the upper-right case of Figure 4), compared to VOCA and MeshTalk; for the even challenging speech parts “waterproof” and “shaving” that need to pout, CodeTalker can produce accurate lip shapes while other methods suffer from the over-smoothing problem and fail to lip-sync correctly (Zoom in for better inspection).

Different from lip movements, facial expressions only have weak correlations with the speech signal, which tends to be static in front of cross-modal mapping ambiguity. To visualize the facial motion dynamics, we calculate the temporal statistics of adjacent-frame facial motions within a sequence. Specifically, we first calculate the inter-frame motion L2 distance and then compute the mean and standard deviation (std) across the sequence at each vertex. The higher mean value indicates stronger facial movements, while the higher std value suggests richer variations of facial dynamics. Two examples are visualized in the last two rows of Figure 4, evidencing that our method outperforms others in achieving both stronger facial movements and a broader range of dynamics. It is mainly attributed to the superiority of the discrete facial motion space, which promotes the robustness to cross-modal uncertainty effectively. Readers are recommended to watch the animation comparisons in the *Supplemental Video*.

Talking style interpolation. Our model can synthesize new speaking styles from the learned style embedding space. To inspect the effects on VOCA-Test, we select two speaking style vectors, *i.e.*, \mathbf{e}_i and \mathbf{e}_j , which correspond to large and slight lip articulations respectively, and interpolate new talking style vectors $\mathbf{s}_{\text{new}} = \mathbf{B} \cdot [\omega \mathbf{e}_i + (1 - \omega) \mathbf{e}_j]$ with a linear coefficient ω . For the synthesized 3D animations of a sampled sequence, we plot the lower-upper lip distances across frames for each style in Figure 5, from which we observe the smooth transition of mouth amplitudes between the two typical styles. It is not only useful to synthesize new talking styles without additional constraints, but also prac-

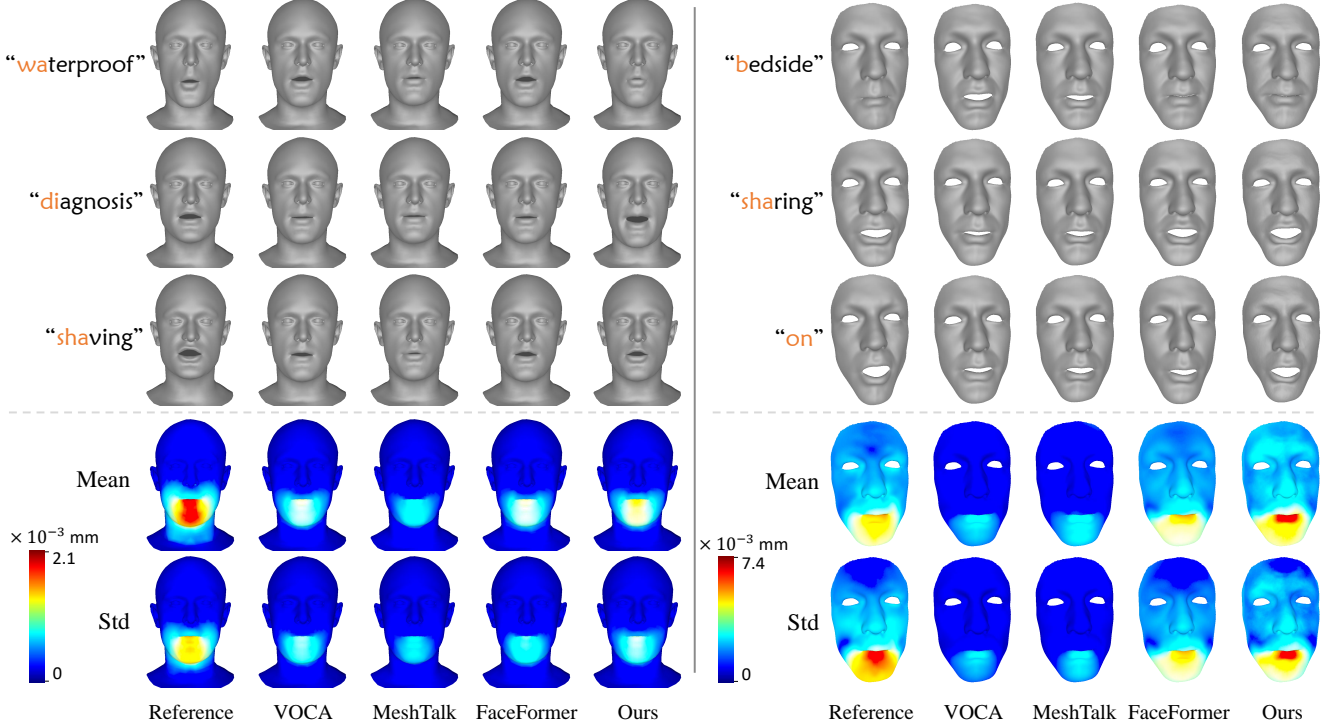


Figure 4. Visual comparisons of sampled facial motions animated by different methods on VOCA-Test (left) and BIWI-Test-B (right). The upper partition shows the facial animation conditioned on different speech parts, while the lower depicts the temporal statistics (mean and standard deviation) of adjacent-frame motion variations within a sequence.

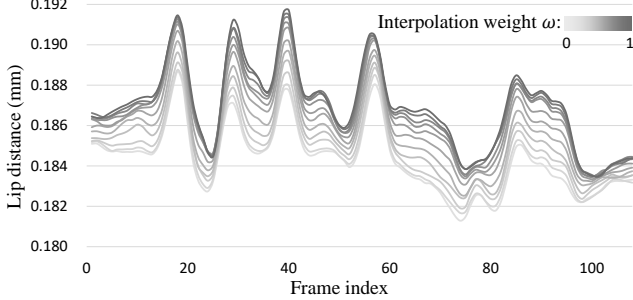


Figure 5. Distance between lower and upper lip for our predictions within a sequence conditioned on different weighted linear combinations of two style vectors.

tical to match a specific speaking performance of an unseen subject during training.

4.4. User Study

The human perception system has been evolutionarily adapted to understanding subtle facial motions and capturing lip synchronization. Thus, it is still the most reliable measure in the speech-driven facial animation task. We conduct a user study to evaluate the quality of animated faces in perceptual lip synchronization and realism, compared with VOCA, MeshTalk, FaceFormer and the ground truth. We adopt A/B tests for each comparison, *i.e.*, ours vs. competitor, in terms of realistic facial animation and lip sync. For

Table 2. User study results on BIWI-Test-B and VOCA-Test. We adopt A/B testing and report the percentage of answers where A is preferred over B.

Competitors	BIWI-Test-B		VOCA-Test	
	Lip Sync	Realism	Lip Sync	Realism
Ours vs. VOCA	92.47	89.25	86.02	84.95
Ours vs. MeshTalk	80.65	82.80	95.70	92.47
Ours vs. FaceFormer	53.76	56.99	70.97	69.89
Ours vs. GT	43.01	49.46	43.01	43.01

BIWI, we obtain the results of four kinds of comparisons by randomly selecting 30 samples from BIWI-Test-B, respectively. To achieve the most variations in terms of speaking styles, we ensure the sampling results can fairly cover all conditioning styles. Thus, 120 A vs. B pairs (30 samples \times 4 comparisons) are created for BIWI-Test-B. Each pair is judged by at least 3 different participants separately, and finally, 372 entries are collected in total. For the user study on VOCASET, we apply the same setting as that on the BIWI dataset, *i.e.*, another 120 A vs. B pairs from VOCA-Test set, finally yielding 372 entries as well. In this study, 31 participants with good vision and hearing ability complete the evaluation successfully. Moreover, each participant is involved in all 8 kinds of comparisons to make better exposure and cover the diversity of favorability.

The percentage of A/B testing in terms of lip sync and re-

Table 3. Ablation study on the representation space of codebook. The performance is measured by the reconstruction error (*i.e.*, L2 error) and lip vertex error on VOCA-Test and BIWI-Test-A.

Variants	VOCA-Test	BIWI-Test-A	
	Rec. Error ($\times 10^{-5}$ mm)	Rec. Error ($\times 10^{-5}$ mm)	Lip Vertex Error ($\times 10^{-4}$ mm)
Shape-ent. codebook	2.75	4.07	6.41
Motion codebook (Ours)	0.08	2.83	4.79

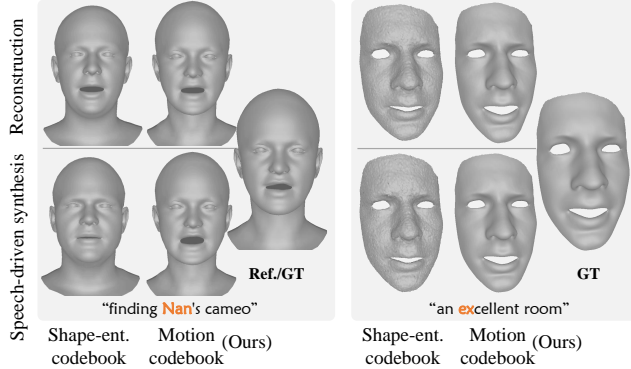


Figure 6. Visual comparisons of reconstruction and speech-driven motion synthesis results with different representation spaces on VOCA-Test (left) and BIWI-Test-A (right).

alism on BIWI-Test-B is tabulated in Table 2, which shows that participants favor CodeTalker over competitors. Based on the visual analysis in Section 4.3, we attribute this to the facial animation synthesized by CodeTalker having more expressive facial motions, accurate lip shape, and well-synchronized mouth movements. For VOCA-Test, which has a nature of fewer upper-face motions, a similar favorability can still be observed in Table 2. We believe the reasons are at least three-fold: subtle motions around the eyes, more accurate lip movements and expressive motions in the lower face. Although be aware of a gap between our predictions and the recorded performance (ground truth), we surprisingly get over 40% preference when compared to the ground truth. Overall, the user study justifies that the facial animations produced by CodeTalker have superior perceptual quality.

4.5. Ablation Studies

We study several key designs of our proposed method in this section, including the representation space and the hyper-parameters of the codebook construction.

Representation space. To study the superiority of our motion-based representation, we construct a baseline that learns a shape-entangled codebook, *i.e.*, the codes represent shapes instead of motions. As shown in Table 3, the baseline decreases the reconstruction accuracy significantly, which is evidenced by the visualized examples in Figure 6. It is mainly because the shape-entangled sequence

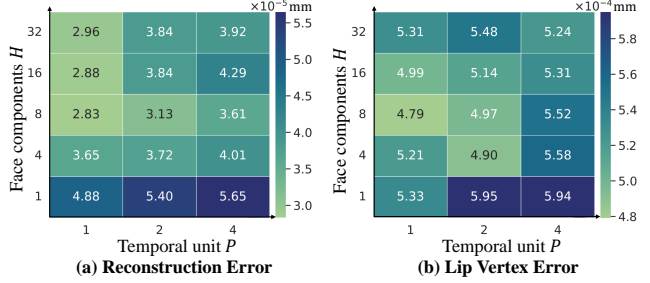


Figure 7. Model performance comparisons with different hyper-parameters of the codebook, *i.e.*, the length of temporal unit P and the number of face components H . We measure the reconstruction error and lip vertex error on BIWI-Test-A.

contains more speaker-specific information that hinders the reusability of the codes. A direct weakness is the poor generalization for self-reconstruction, which further impedes cross-modal mapping correctness. In contrast, our proposed speaker-agnostic motion representation is more effective to represent generic motion priors shared across individuals, and hence promotes the quality of both the self-reconstruction and the speech-driven motion synthesis.

Codebook construction. We further study the hyper-parameters used for codebook construction. We evaluate the performance of different settings $\langle P, H \rangle$ by measuring their reconstruction accuracy and cross-modal mapping accuracy (namely lip vertex error). First, we evaluate the reconstruction accuracy as shown in Figure 7(a). On one hand, increasing P degrades the reconstruction accuracy, which could be explained by the increased complexity of the motion to be represented. On the other hand, increasing H eases the reconstruction but risks over-fitting, which explains the general benefits ($H < 8$) but inferior performance when $H \geq 8$. Notably, a similar trend could be found in the cross-modal mapping performance, as shown in Figure 7(b). We conjecture that complex motion primitives cause lower reusability and higher redundancy, resulting in ambiguity in the cross-modal code query process.

5. Discussion and Conclusion

We demonstrated the advantages of casting speech-driven facial animation as a code query task in the discrete space, which notably promotes the motion synthesis quality against cross-modal ambiguity. By comparing to the existing state-of-the-arts, our proposed method shows superiority in achieving accurate lip sync and vivid facial expressions. However, we still follow the assumption that facial motions are independent of shapes, whose rationality may deserve further studies. Also, the overall perceptual quality still lags behind the ground truth, mainly because of the limited paired audio-visual data. As a future work, it is interesting to guide the 3D facial animation by utilizing priors from large-scale available talking head videos.

References

- [1] Mohammed M Alghamdi, He Wang, Andrew J Bulpitt, and David C Hogg. Talking head from speech audio using a pre-trained image generator. In *ACM International Conference on Multimedia (MM)*, 2022. 2
- [2] Tenglong Ao, Qingzhe Gao, Yuke Lou, Baoquan Chen, and Libin Liu. Rhythmic gesticulator: Rhythm-aware co-speech gesture synthesis with hierarchical neural embeddings. *ACM Transactions on Graphics (TOG)*, pages 1–19, 2022. 3
- [3] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2020. 4
- [4] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013. 3
- [5] Chen Cao, Hongzhi Wu, Yanlin Weng, Tianjia Shao, and Kun Zhou. Real-time facial animation with image-based dynamic avatars. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016. 2
- [6] Yong Cao, Wen C Tien, Petros Faloutsos, and Frédéric Pighin. Expressive speech-driven facial animation. *ACM Transactions on Graphics (TOG)*, 24(4):1283–1302, 2005. 2
- [7] Lele Chen, Guofeng Cui, Celong Liu, Zhong Li, Ziyi Kou, Yi Xu, and Chenliang Xu. Talking-head generation with rhythmic head motion. In *European Conference on Computer Vision (ECCV)*, 2020. 2
- [8] Lele Chen, Zhiheng Li, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. Lip movements generation at a glance. In *European Conference on Computer Vision (ECCV)*, 2018. 2
- [9] Joon Son Chung and Andrew Zisserman. Out of time: automated lip sync in the wild. In *Asian Conference on Computer Vision (ACCV)*, 2016. 2
- [10] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J Black. Capture, learning, and synthesis of 3d speaking styles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5, 14
- [11] Dipanjan Das, Sandika Biswas, Sanjana Sinha, and Brojeshwar Bhowmick. Speech-driven facial animation using cascaded gans for learning of motion and texture. In *European Conference on Computer Vision (ECCV)*, 2020. 2
- [12] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: an animator-centric viseme model for expressive lip synchronization. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 2
- [13] Paul Ekman and Wallace V Friesen. Facial action coding system. *Environmental Psychology & Nonverbal Behavior*, 1978. 2
- [14] Michael Elad and Michal Aharon. Image denoising via learned dictionaries and sparse representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006. 2
- [15] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [16] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. Faceformer: Speech-driven 3d facial animation with transformers. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 4, 5, 6, 14
- [17] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. Joint audio-text model for expressive speech-driven 3d facial animation. In *ACM Symposium on Interactive 3D Graphics and Games (i3D)*, 2022. 2
- [18] Gabriele Fanelli, Juergen Gall, Harald Romsdorfer, Thibaut Weise, and Luc Van Gool. A 3-d audio-visual corpus of affective communication. *IEEE Transactions on Multimedia (TMM)*, 12(6):591–598, 2010. 5
- [19] Cletus G Fisher. Confusions among visually perceived consonants. *Journal of speech and hearing research*, 11(4):796–804, 1968. 2
- [20] Ohad Fried, Ayush Tewari, Michael Zollhöfer, Adam Finkelstein, Eli Shechtman, Dan B Goldman, Kyle Genova, Zeyu Jin, Christian Theobalt, and Maneesh Agrawala. Text-based editing of talking-head video. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 2
- [21] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
- [22] Shuhang Gu, Wangmeng Zuo, Qi Xie, Deyu Meng, Xi-angchu Feng, and Lei Zhang. Convolutional sparse coding for image super-resolution. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015. 2
- [23] Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2
- [24] Ikhsanul Habibie, Weipeng Xu, Dushyant Mehta, Lingjie Liu, Hans-Peter Seidel, Gerard Pons-Moll, Mohamed Elgharib, and Christian Theobalt. Learning speech-driven 3d conversational gestures from video. In *ACM International Conference on Intelligent Virtual Agents (IVA)*, 2021. 2
- [25] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017. 12
- [26] Zhitong Huang, Nanxuan Zhao, and Jing Liao. Unicolor: A unified framework for multi-modal colorization with transformer. *ACM Transactions on Graphics (TOG)*, pages 1–16, 2022. 2
- [27] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015. 12
- [28] Xinya Ji, Hang Zhou, Kaisiyuan Wang, Qianyi Wu, Wayne Wu, Feng Xu, and Xun Cao. Eammn: One-shot emotional talking face via audio-based emotion-aware motion model. In *ACM SIGGRAPH Conference*, 2022. 2

- [29] Xinya Ji, Hang Zhou, Kaisiyuan Wang, Wayne Wu, Chen Change Loy, Xun Cao, and Feng Xu. Audio-driven emotional video portraits. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [30] Younghyun Jo and Seon Joo Kim. Practical single-image super-resolution using look-up table. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [31] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)*, 36(4):1–12, 2017. 1, 2, 14
- [32] Hyeonwoo Kim, Pablo Garrido, Ayush Tewari, Weipeng Xu, Justus Thies, Matthias Niessner, Patrick Pérez, Christian Richardt, Michael Zollhöfer, and Christian Theobalt. Deep video portraits. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018. 2
- [33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2014. 5
- [34] Avisek Lahiri, Vivek Kwatra, Christian Frueh, John Lewis, and Chris Bregler. Lipsync3d: Data-efficient learning of personalized 3d talking faces from video using pose and lighting normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [35] John Lewis. Automated lip-sync: Background and techniques. *The Journal of Visualization and Computer Animation*, 2(4):118–122, 1991. 2
- [36] Hao Li, Jihun Yu, Yuting Ye, and Chris Bregler. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics (TOG)*, 32(4):42–1, 2013. 2
- [37] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Transactions on Graphics (TOG)*, 36(6):194–1, 2017. 1, 5
- [38] Borong Liang, Yan Pan, Zhizhi Guo, Hang Zhou, Zhibin Hong, Xiaoguang Han, Junyu Han, Jingtuo Liu, Errui Ding, and Jingdong Wang. Expressive talking head generation with granular audio-visual control. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2
- [39] Xian Liu, Yinghao Xu, Qianyi Wu, Hang Zhou, Wayne Wu, and Bolei Zhou. Semantic-aware implicit neural audio-driven video portrait generation. In *European Conference on Computer Vision (ECCV)*, 2022. 2
- [40] Yilong Liu, Feng Xu, Jinxiang Chai, Xin Tong, Lijuan Wang, and Qiang Huo. Video-audio driven real-time facial animation. *ACM Transactions on Graphics (TOG)*, 34(6):1–10, 2015. 2
- [41] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019. 5
- [42] DW Massaro, MM Cohen, M Tabain, J Beskow, and R Clark. Animated speech: research progress and applications. *Audiovisual Speech Processing*, pages 309–345, 2012. 2
- [43] Wesley Matthysen and Werner Verhelst. Audiovisual speech synthesis: An overview of the state-of-the-art. *Speech Communication*, 66:182–217, 2015. 2
- [44] Evonne Ng, Hanbyul Joo, Liwen Hu, Hao Li, Trevor Darrell, Angjoo Kanazawa, and Shiry Ginosar. Learning to listen: Modeling non-deterministic dyadic facial motion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 4, 13
- [45] Frederic I Parke and Keith Waters. *Computer facial animation*. CRC press, 2008. 2
- [46] Jialun Peng, Dong Liu, Songcen Xu, and Houqiang Li. Generating diverse structure for image inpainting with hierarchical vq-vae. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [47] Hai Xuan Pham, Yuting Wang, and Vladimir Pavlovic. End-to-end learning for 3d facial animation from speech. In *ACM International Conference on Multimodal Interaction (ICMI)*, 2018. 2
- [48] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Namboodiri, and CV Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *ACM International Conference on Multimedia (MM)*, 2020. 2
- [49] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019. 2
- [50] Alexander Richard, Colin Lea, Shugao Ma, Jurgen Gall, Fernando De la Torre, and Yaser Sheikh. Audio-and gaze-driven facial animation of codec avatars. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021. 1
- [51] Alexander Richard, Michael Zollhöfer, Yandong Wen, Fernando De la Torre, and Yaser Sheikh. Meshtalk: 3d face animation from speech using cross-modality disentanglement. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 5, 6, 14
- [52] Shuai Shen, Wanhua Li, Zheng Zhu, Yueqi Duan, Jie Zhou, and Jiwen Lu. Learning dynamic facial radiance fields for few-shot talking head synthesis. In *European Conference on Computer Vision (ECCV)*, 2022. 2
- [53] Sarah Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica Hodgins, and Iain Matthews. A deep learning approach for generalized speech animation. *ACM Transactions on Graphics (TOG)*, 36(4):1–11, 2017. 2
- [54] Sarah L Taylor, Moshe Mahler, Barry-John Theobald, and Iain Matthews. Dynamic units of visual speech. In *ACM/Eurographics Symposium on Computer Animation (SCA)*, 2012. 1, 2
- [55] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. In *European Conference on Computer Vision (ECCV)*, 2020. 2
- [56] Radu Timofte, Vincent De Smet, and Luc Van Gool. Anchored neighborhood regression for fast example-based super-resolution. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013. 2
- [57] Radu Timofte, Vincent De Smet, and Luc Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision (ACCV)*, 2014. 2

- [58] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 12
- [59] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 12
- [60] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 1, 2, 3
- [61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 2
- [62] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *International Journal of Computer Vision (IJCV)*, 128(5):1398–1413, 2020. 2
- [63] Qianyun Wang, Zhenfeng Fan, and Shihong Xia. 3d-talkemo: Learning to synthesize 3d emotional talking head. *arXiv preprint arXiv:2104.12051*, 2021. 2
- [64] Suzhen Wang, Lincheng Li, Yu Ding, and Xin Yu. One-shot talking face generation from single-speaker audio-visual correlation learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2022. 2
- [65] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics (TOG)*, 30(4):1–10, 2011. 2
- [66] Yuyu Xu, Andrew W Feng, Stacy Marsella, and Ari Shapiro. A practical and configurable lip sync method for games. In *Proceedings of Motion on Games*, 2013. 1, 2
- [67] Karren Yang, Dejan Marković, Steven Krenn, Vasu Agrawal, and Alexander Richard. Audio-visual speech codecs: Rethinking audio-visual speech enhancement by re-synthesis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3
- [68] Ran Yi, Zipeng Ye, Juyong Zhang, Hujun Bao, and Yong-Jin Liu. Audio-driven talking face video generation with learning-based personalized head pose. *arXiv preprint arXiv:2002.10137*, 2020. 2
- [69] Hang Zhou, Yasheng Sun, Wayne Wu, Chen Change Loy, Xiaogang Wang, and Ziwei Liu. Pose-controllable talking face generation by implicitly modularized audio-visual representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2
- [70] Shangchen Zhou, Kelvin CK Chan, Chongyi Li, and Chen Change Loy. Towards robust blind face restoration with codebook lookup transformer. *Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 13
- [71] Yang Zhou, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh. Visemenet: Audio-driven animator-centric speech animation. *ACM Transactions on Graphics (TOG)*, 37(4):1–10, 2018. 2
- [72] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler, Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. State of the art on monocular 3d face reconstruction, tracking, and applications. *Computer Graphics Forum (CGF)*, 37(2):523–550, 2018. 2

Appendix

This supplemental document contains four sections: Section A shows implementation details of our CodeTalker; Section B presents more discussions on the proposed method; Section C presents details of the user study; and Section D presents short descriptions of the supplemental video. The source code and trained model will also be released upon publication.

A. Implementation Details

A.1. Hyper-parameters of Codebook

We have explored and discussed the important hyper-parameters of our motion codebook in Section 4.5 “Codebook construction” on the BIWI dataset in the main paper. Here we provide more specific parameters adopted for CodeTalker trained on the two datasets. For BIWI, we have the ground truth for quantitative evaluation on the testing set BIWI-Test-A to determine a group of parameters $P = 1$ and $H = 8$ for high-quality results (*i.e.*, Section 4.5 “Codebook construction” in the main paper). Additionally, we set the codebook item number $N = 256$ and the dimension of items $C = 128$. Although more codebook items and dimensions could ease reconstruction, the redundant elements may cause ambiguity in speech-driven motion synthesis. Hence, we did not heavily tune these parameters and just empirically set them for good visual quality. For VOCASET, since there is no ground truth for us to obtain the quantitative results, we empirically select a group of parameters (*i.e.*, $N = 256$, $P = 1$, $H = 16$, $C = 64$), which could produce visually plausible facial animations in our experiments.

A.2. Network Architecture

To improve the reproducibility of our CodeTalker, we further illustrate the detailed network architectures for the facial motion space learning and the speech-driven motion synthesis (Section 3.1 and 3.2 in the main paper, respectively), which are shown Table 4.

B. More Discussions on CodeTalker

B.1. Instance Normalization in Self-reconstruction Learning

Instance Normalization [58] (IN) has been widely used in the field of style transfer [25, 59], which is defined as:

$$\text{IN}(x) = \gamma \left(\frac{x - \mu(x)}{\sigma(x)} \right) + \beta. \quad (8)$$

Different from BN [27] layers, here $\mu(x)$ and $\sigma(x)$ are computed across temporal dimensions independently for each

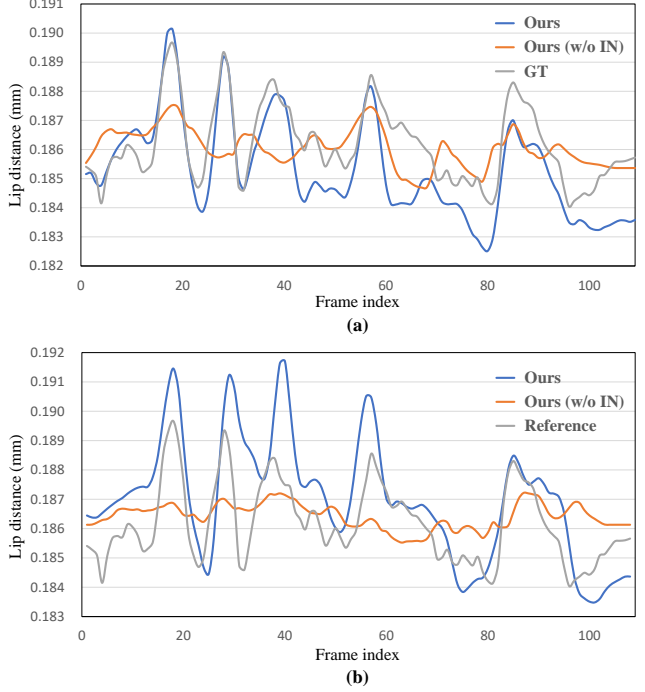


Figure 8. Distance between lower and upper lip within a sampled sequence from VOCA-Test of (a) reconstruction and (b) speech-driven motion synthesis results produced by different variants.

channel within each sample:

$$\mu_{nc}(x) = \frac{1}{T} \sum_{t=1}^T x_{nct} \quad (9)$$

$$\sigma_{nc}(x) = \sqrt{\frac{1}{T} \sum_{t=1}^T (x_{nct} - \mu_{nc}(x))^2 + \epsilon} \quad (10)$$

Interestingly, we empirically find that normalizing feature statistics (*i.e.*, mean and variance) with IN (not BN due to small mini-batch size) can boost the performance of our CodeTalker in self-reconstruction learning, as shown in Table 5. In addition, it can also make self-reconstruction training more stable. To better show the gain of normalization, we also visualize the lip distance of a sampled sequence of reconstruction results from VOCA-Test in Figure 8(a). The visualization result indicates that the predicted lip amplitudes are closer to those of the ground truth by equipping with IN, while the ablated variant (*i.e.*, Ours (w/o IN)) cannot reconstruct lip movements with accurate amplitudes. The speech-driven facial motion synthesis (stage two) can also benefit from the facial motion codebook learned in self-reconstruction with IN, as shown in Figure 8(b). Note that we synthesize facial motions conditioned on a randomly sampled speaking style. We conjecture that facial motions with different magnitudes could be well encapsulated into

Table 4. Parameter illustration of network architectures. $C(k,s,p,n)$ denotes a 1D Convolutional layer with kernel size k , stride size s , padding size p , and output channels of n . $T_{enc}(d1,d2,h,l)$ denotes a transformer encoder layer with basic channel number of $d1$, forward channel number of $d2$, self-attention head number of h , and layer number l , while similarly, T_{dec} represents a transformer decoder layer. $L(n)$ denotes a linear layer with output channels of n . $CA[\cdot]$ stands for the additional cross-attention input for transformer decoders. $l_{CM} = 12$ for BIWI, while $l_{CM} = 6$ for VOCASET. $n \cdot T$ stands for the interpolated audio feature length in order to align with visual frames, where $n = 2$ for BIWI and $n = 1$ for VOCASET. ‘+’ denotes the channel-wise addition. “Drop” means the dropout operation.

Stage	Module	Input \rightarrow Output	Layer Operation
I	Encoder	$\mathbf{M}(T, V, 3) \rightarrow \mathbf{M}(T, V \cdot 3)$	Reshape
		$\mathbf{M}(T, V \cdot 3) \rightarrow \mathbf{Z}_e^1(T, 1024)$	$L(1024) \rightarrow LReLU \rightarrow C(5,1,2,1024) \rightarrow LReLU \rightarrow IN$
		$\mathbf{Z}_e^1(T, 1024) \rightarrow \mathbf{Z}_e^2(T, H \cdot C)$	$L(1024) \rightarrow T_{enc}(1024,1536,8,6) \rightarrow L(H \cdot C)$
		$\mathbf{Z}_e^2(T, H \cdot C) \rightarrow \mathbf{Z}_q(T, H, C)$	Reshape \rightarrow Quantize
	Decoder	$\mathbf{Z}_q(T, H, C) \rightarrow \mathbf{Z}_q(T, H \cdot C)$	Reshape
		$\mathbf{Z}_q(T, H \cdot C) \rightarrow \mathbf{Z}_d^1(T, 1024)$	$L(1024) \rightarrow C(5,1,2,1024) \rightarrow LReLU \rightarrow IN$
		$\mathbf{Z}_d^1(T, 1024) \rightarrow \hat{\mathbf{M}}(T, V \cdot 3)$	$L(1024) \rightarrow T_{enc}(1024,1536,8,6) \rightarrow L(V \cdot 3)$
II	Speech Encoder	$\mathbf{A}(T, d) \rightarrow \mathbf{F}_e^1(T', 512)$	$C(10,5,0,512) \rightarrow GN \rightarrow GeLU \rightarrow C(3,2,0,512) \rightarrow GN \rightarrow GeLU$
		$\mathbf{F}_e^1(T', 512) \rightarrow \mathbf{F}_e^2(n \cdot T, 768)$	$\rightarrow C(3,2,0,512) \rightarrow GN \rightarrow GeLU \rightarrow C(3,2,0,512) \rightarrow GN \rightarrow GeLU$
		$\mathbf{F}_e^2(n \cdot T, 768) \rightarrow \mathbf{F}_e^3(n \cdot T, 1024)$	$\rightarrow C(3,2,0,512) \rightarrow GN \rightarrow GeLU \rightarrow C(3,2,0,512) \rightarrow GN \rightarrow GeLU$
			$\rightarrow C(2,2,0,512) \rightarrow GN \rightarrow GeLU \rightarrow C(2,2,0,512) \rightarrow GN \rightarrow GeLU$
	Cross-modal Decoder	$\hat{\mathbf{M}}_{past}(T, V \cdot 3) \rightarrow \mathbf{F}_{emb}^{past}(T, 1024)$	Interpolate $\rightarrow LN \rightarrow L(768) \rightarrow Drop$
		$\mathbf{F}_{emb}^{past}(T, 1024) \rightarrow \hat{\mathbf{Z}}_d^1(T, 1024)$	$T_{enc}(768,3072,12,12) \rightarrow L(1024)$
		$\hat{\mathbf{Z}}_d^1(T, 1024) \rightarrow \hat{\mathbf{Z}}_q(T, H, C)$	$L(1024) \rightarrow +StyleVector$
		$\hat{\mathbf{Z}}_q(T, H, C) \rightarrow \hat{\mathbf{Z}}_q(T, H \cdot C)$	$T_{dec}(1024,2048,4,l_{CM})$ with $CA[\mathbf{F}_e^3] \rightarrow L(H \cdot C)$
		$\hat{\mathbf{Z}}_q(T, H \cdot C) \rightarrow \hat{\mathbf{Z}}_q(T, H \cdot C)$	Reshape \rightarrow Quantize
		$\hat{\mathbf{Z}}_q(T, H \cdot C) \rightarrow \hat{\mathbf{Z}}_d^2(T, 1024)$	Reshape
		$\hat{\mathbf{Z}}_d^2(T, 1024) \rightarrow \hat{\mathbf{M}}(T, V \cdot 3)$	$L(1024) \rightarrow C(5,1,2,1024) \rightarrow LReLU \rightarrow IN$
			$L(1024) \rightarrow T_{enc}(1024,1536,8,6) \rightarrow L(V \cdot 3)$

Table 5. Ablation study on the Instance Normalization (IN) for self-reconstruction learning. The performance is measured by the reconstruction error on VOCA-Test and BIWI-Test-A.

Variants	Reconstruction Error	
	VOCA-Test ($\times 10^{-5}$ mm)	BIWI-Test-A ($\times 10^{-5}$ mm)
Ours (w/o IN)	0.12	3.27
Ours	0.08	2.83

the discrete motion prior by normalizing temporal elements within each channel. The rationality and effect of IN deserve further studies as our potential direction.

B.2. Alternative Data Flow and Supervision

As we have summarized in Section 2.2 of the main paper, recent works explore the power of discrete prior learning in a large variety of tasks, among which most existing Vector Quantization (VQ)-based works [44, 70] adopt categorical cross-entropy (CE) loss to supervise their token predictions.

Table 6. Comparison of lip-sync errors. We compare different methods on BIWI-Test-A. Lower means better. λ is the weighting factor.

Method	Lip Vertex Error ($\times 10^{-4}$ mm)
Alter. (\mathcal{L}_{ce})	9.6356
Alter. ($\lambda \mathcal{L}_{ce} + \mathcal{L}_{reg}$)	5.1138
Alter. ($\lambda \mathcal{L}_{ce} + \mathcal{L}_{reg} + \mathcal{L}_{motion}$)	5.0254
CodeTalker (Ours)	4.7914

Hence, we also explore some alternative data flow and supervision frameworks as our cross-modal decoder, which is shown in Figure 9. It is worth noting that the style vector and audio features are omitted for simplicity.

Different from our cross-modal decoder in the main paper, the alternative takes past motion code as input and then autoregressively predicts code sequences in form of n-way classification. The predicted code sequence then retrieves the respective code items from the learned codebook \mathcal{Z} , and

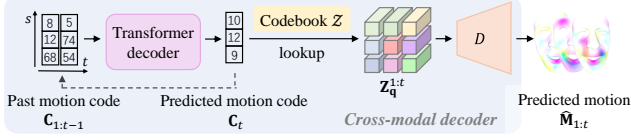


Figure 9. Alternative data flow and supervision framework of our cross-modal decoder. Note that we omit the style vector and audio features input for simplicity. Given the past motion code as input, the alternative cross-modal decoder first autoregressively predict motion code and then decode them into motions with the pre-trained codebook and decoder.

further produces facial motion sequences through the fixed decoder D . A CE loss is adopted to penalize error between the predicted code sequence $\hat{\mathbf{c}} \in \{0, \dots, |N| - 1\}^{T' \cdot H}$ and the ground truth \mathbf{c} generated by the pre-trained encoder E :

$$\mathcal{L}_{ce} = \sum_{i=0}^{T' \cdot H} -\mathbf{c}_i \log(\hat{\mathbf{c}}_i). \quad (11)$$

We train the alternatives with the same settings as those in the main paper (Section 3.3). The lip-sync evaluation result is tabulated in Table 6. Alternative model with \mathcal{L}_{ce} alone cannot converge well due to the difficult cross-modality mapping of token prediction. While adding more constraints (*i.e.*, \mathcal{L}_{reg} and \mathcal{L}_{motion} in the main paper Eq. 6) can ease the difficulty of token prediction learning, the performance is still limited with this token prediction framework. Overall, the lower average lip error achieved by our CodeTalker suggests its framework superiority in terms of the accuracy of lip movements.

C. User Study

The designed user study interface is shown in Figure 10. A user study is expected to be completed with 5–10 minutes (24 video pairs \times 5 seconds \times 3 times watching). To remove the impact of random selection, we filter out those comparison results completed in less than two minutes. For each participant, the user study interface shows 24 video pairs and the participant is instructed to judge the videos twice with the following two questions, respectively: “Comparing the lips of two faces, which one is more in sync with the audio?” and “Comparing the two full faces, which one looks more realistic?”.

D. Video Comparison

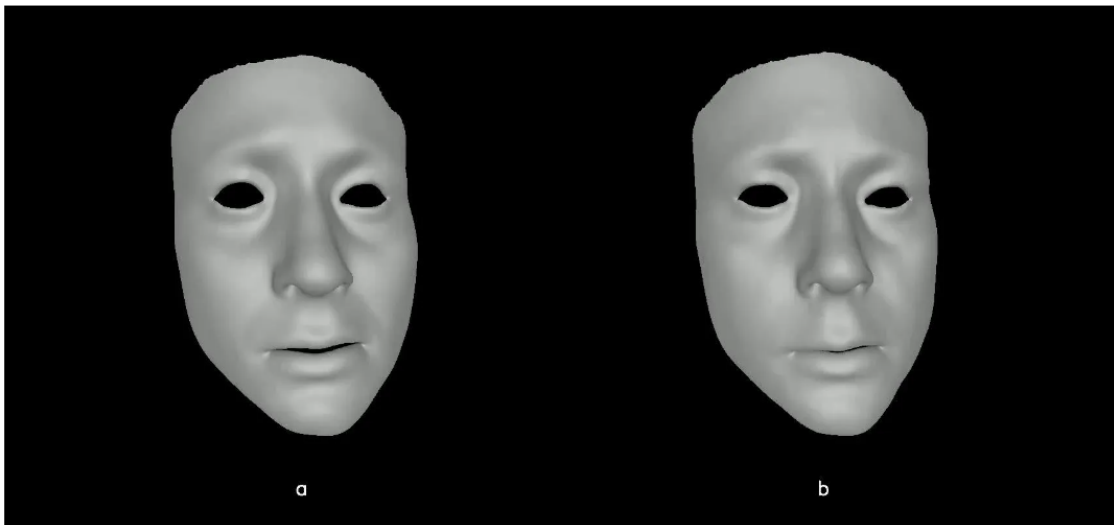
To better evaluate the qualitative results produced by competitors [10, 16, 31, 51] and our CodeTalker, we provide a supplemental video for demonstration and comparison. Specifically, we test our model using various audio clips, including the audio clips extracted from TED and TEDx videos, audio sequences from the VOCASET and

BIWI datasets, and the speech from supplementary videos of previous methods. The video shows that CodeTalker can synthesize natural and plausible facial animations with well-synchronized lip movements. It is worth noting that, compared to the competitors (*i.e.*, VOCA, MeshTalk and FaceFormer) suffering from the over-smoothing problem, our CodeTalker can produce more vivid and realistic facial motions and better lip sync. Besides, we also show the talking style interpolation results and facial animations of talking in different languages.

Instructions:

Please watch the short videos (duration ~5s) of two animated talking heads. You need to choose the talking head (a or b) that moves **more naturally** in terms of the **full face and the lips (two questions for each video)**. The total duration for this survey is about 5-10mins.

Reminder: Please turn on the **sound** on your computer when watching.



1.1 Comparing the lips of two faces, which one is more in sync with the audio?

- ☐ a
- ☐ b

1.2 Comparing the two full faces, which one looks more realistic?

- ☐ a
- ☐ b

Figure 10. Designed user study interface. Each participant need to answer 24 video pairs and here only one video pair is shown due to the page limit.