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

Jinbo Xing[1] Menghan Xia[2,*] Yuechen Zhang[1] Xiaodong Cun[2] Jue Wang[2] Tien-Tsin Wong[1]
[1]The Chinese University of Hong Kong [2]Tencent AI Lab
{jbxing,yczhang21,ttwong}@cse.cuhk.edu.hk {menghanxyz,vinthony,arphid}@gmail.com

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.

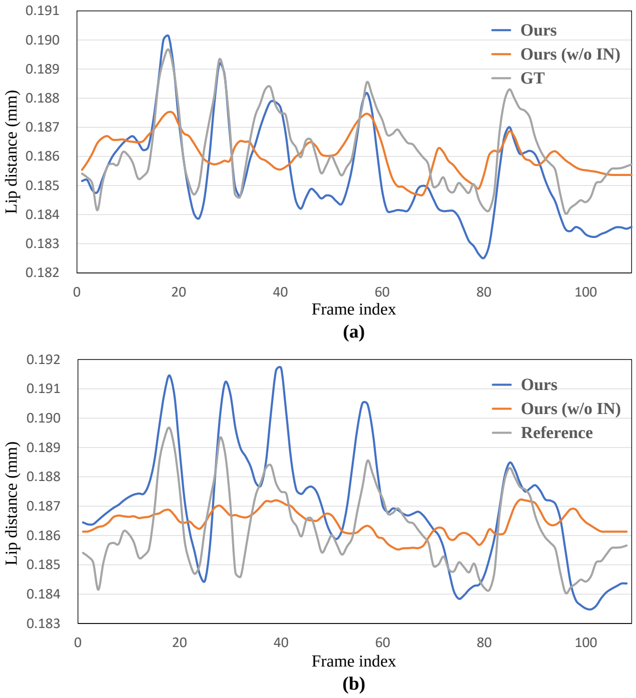
A. Implementation Details

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 1.



(a)

(b)

Figure 1. 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.

B. More Discussions on CodeTalker

B.1. Instance Normalization in Self-reconstruction Learning

Instance Normalization [8] (IN) has been widely used in the filed of style transfer [3, 9], which is defined as:

$$\mathrm{IN}(x) = \gamma(\frac{x - \mu(x)}{\sigma(x)}) + \beta. \quad (1)$$

Different from BN [4] layers, here $\mu(x)$ and $\sigma(x)$ are computed across temporal dimensions independently for each

[*] Corresponding Author.

Table 1. 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. $\mathrm{T}_{\mathrm{enc}}$(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, $\mathrm{T}_{\mathrm{dec}}$ represents a transformer decoder layer. L(n) denotes a linear layer with output channels of n. CA[·] stands for the additional cross-attention input for transformer decoders. $\mathrm{l}_{\mathrm{CM}} = 12$ for BIWI, while $\mathrm{l}_{\mathrm{CM}} = 6$ 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 → 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) → LReLU → C(5,1,2,1024) → LReLU → IN
		$\mathbf{Z}_e^1(T, 1024) \rightarrow \mathbf{Z}_e^2(T, H \cdot C)$	L(1024) → $\mathrm{T}_{\mathrm{enc}}$(1024,1536,8,6) → L($H \cdot C$)
		$\mathbf{Z}_e^2(T, H \cdot C) \rightarrow \mathbf{Z}_{\mathbf{q}}(T, H, C)$	Reshape → Quantize
	Decoder	$\mathbf{Z}_{\mathbf{q}}(T, H, C) \rightarrow \mathbf{Z}_{\mathbf{q}}(T, H \cdot C)$	Reshape
		$\mathbf{Z}_{\mathbf{q}}(T, H \cdot C) \rightarrow \mathbf{Z}_d^1(T, 1024)$	L(1024) → C(5,1,2,1024) → LReLU → IN
		$\mathbf{Z}_d^1(T, 1024) \rightarrow \hat{\mathbf{M}}(T, V \cdot 3)$	L(1024) → $\mathrm{T}_{\mathrm{enc}}$(1024,1536,8,6) → L($V \cdot 3$)
II	Speech Encoder	$\mathbf{A}(T, d) \rightarrow \mathbf{F}_e^1(T', 512)$	C(10,5,0,512) → GN → GeLU → C(3,2,0,512) → GN → GeLU → C(3,2,0,512) → GN → GeLU → C(3,2,0,512) → GN → GeLU → C(3,2,0,512) → GN → GeLU → C(3,2,0,512) → GN → GeLU → C(2,2,0,512) → GN → GeLU → C(2,2,0,512) → GN → GeLU
		$\mathbf{F}_e^1(T', 512) \rightarrow \mathbf{F}_e^2(n \cdot T, 768)$	Interpolate → LN → L(768) → Drop
		$\mathbf{F}_e^2(n \cdot T, 768) \rightarrow \mathbf{F}_e^3(n \cdot T, 1024)$	$\mathrm{T}_{\mathrm{enc}}$(768,3072,12,12) → L(1024)
	Cross-modal Decoder	$\hat{\mathbf{M}}_{\text{past}}(T, V \cdot 3) \rightarrow \mathbf{F}_{\text{emb}}^{\text{past}}(T, 1024)$	L(1024) → +StyleVector
		$\mathbf{F}_{\text{emb}}^{\text{past}}(T, 1024) \rightarrow \hat{\mathbf{Z}}_d^1(T, 1024)$	$\mathrm{T}_{\mathrm{dec}}$(1024,2048,4,$\mathrm{l}_{\mathrm{CM}}$) with CA[$\mathbf{F}_e^3$] → L($H \cdot C$)
		$\hat{\mathbf{Z}}_d^1(T, H \cdot C) \rightarrow \hat{\mathbf{Z}}_{\mathbf{q}}(T, H, C)$	Reshape → Quantize
		$\hat{\mathbf{Z}}_{\mathbf{q}}(T, H, C) \rightarrow \hat{\mathbf{Z}}_{\mathbf{q}}(T, H \cdot C)$	Reshape
		$\hat{\mathbf{Z}}_{\mathbf{q}}(T, H \cdot C) \rightarrow \hat{\mathbf{Z}}_d^2(T, 1024)$	L(1024) → C(5,1,2,1024) → LReLU → IN
		$\hat{\mathbf{Z}}_d^2(T, 1024) \rightarrow \hat{\mathbf{M}}(T, V \cdot 3)$	L(1024) → $\mathrm{T}_{\mathrm{enc}}$(1024,1536,8,6) → L($V \cdot 3$)

Table 2. 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**

channel within each sample:

$$\mu_{nc}(x) = \frac{1}{T} \sum_{t=1}^{T} x_{nct} \tag{2}$$

$$\sigma_{nc}(x) = \sqrt{\frac{1}{T} \sum_{t=1}^{T} (x_{nct} - \mu_{nc}(x))^2 + \epsilon} \tag{3}$$

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 2. 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 1(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 1(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 the discrete motion prior by normalizing temporal elements within each channel. The rationality and effect of IN deserve further studies as our potential direction.

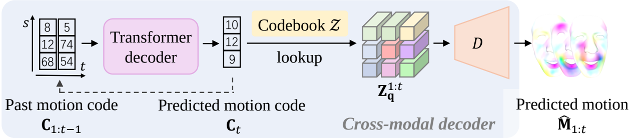


Figure 2. Alternative data flow and supervision framework of our cross-modal decoder. Note that we omit the style vector and audio features input for simplicity. Given the past motion code as input, the alternative cross-modal decoder first autoregressively predict motion code and then decode them into motions with the pre-trained codebook and decoder.

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 [6,10] adopt categorical cross-entropy (CE) loss to supervise their token predictions. Hence, we also explore some alternative data flow and supervision frameworks as our cross-modal decoder, which is shown in Figure 2. 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 further produces facial motion sequences through the fixed decoder D. A CE loss is adopted to penalize error between the predicted code sequence $\hat{\mathbf{c}} \in \{0, \ldots, |N|-1\}^{T' \cdot H}$ and the ground truth $\mathbf{c}$ generated by the pre-trained encoder E:

$$\mathcal{L}_{\text{ce}} = \sum_{i=0}^{T' \cdot H} -\mathbf{c}_i \log(\hat{\mathbf{c}}_i). \tag{4}$$

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 3. Alternative model with $\mathcal{L}_{\text{ce}}$ alone cannot converge well due to the difficult cross-modality mapping of token prediction. While adding more constraints (*i.e.*, $\mathcal{L}_{\text{reg}}$ and $\mathcal{L}_{\text{motion}}$ 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

Table 3. 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. ($\mathcal{L}_{\text{ce}}$)	9.6356
Alter. ($\lambda\mathcal{L}_{\text{ce}}$+$\mathcal{L}_{\text{reg}}$)	5.1138
Alter. ($\lambda\mathcal{L}_{\text{ce}}$+$\mathcal{L}_{\text{reg}}$+$\mathcal{L}_{\text{motion}}$)	5.0254
CodeTalker (Ours)	**4.7914**

The designed user study interface is shown in Figure 3. A user study is expected to be completed with 5–10 minutes (24 video pairs × 5 seconds × 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 [1, 2, 5, 7] 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.

References

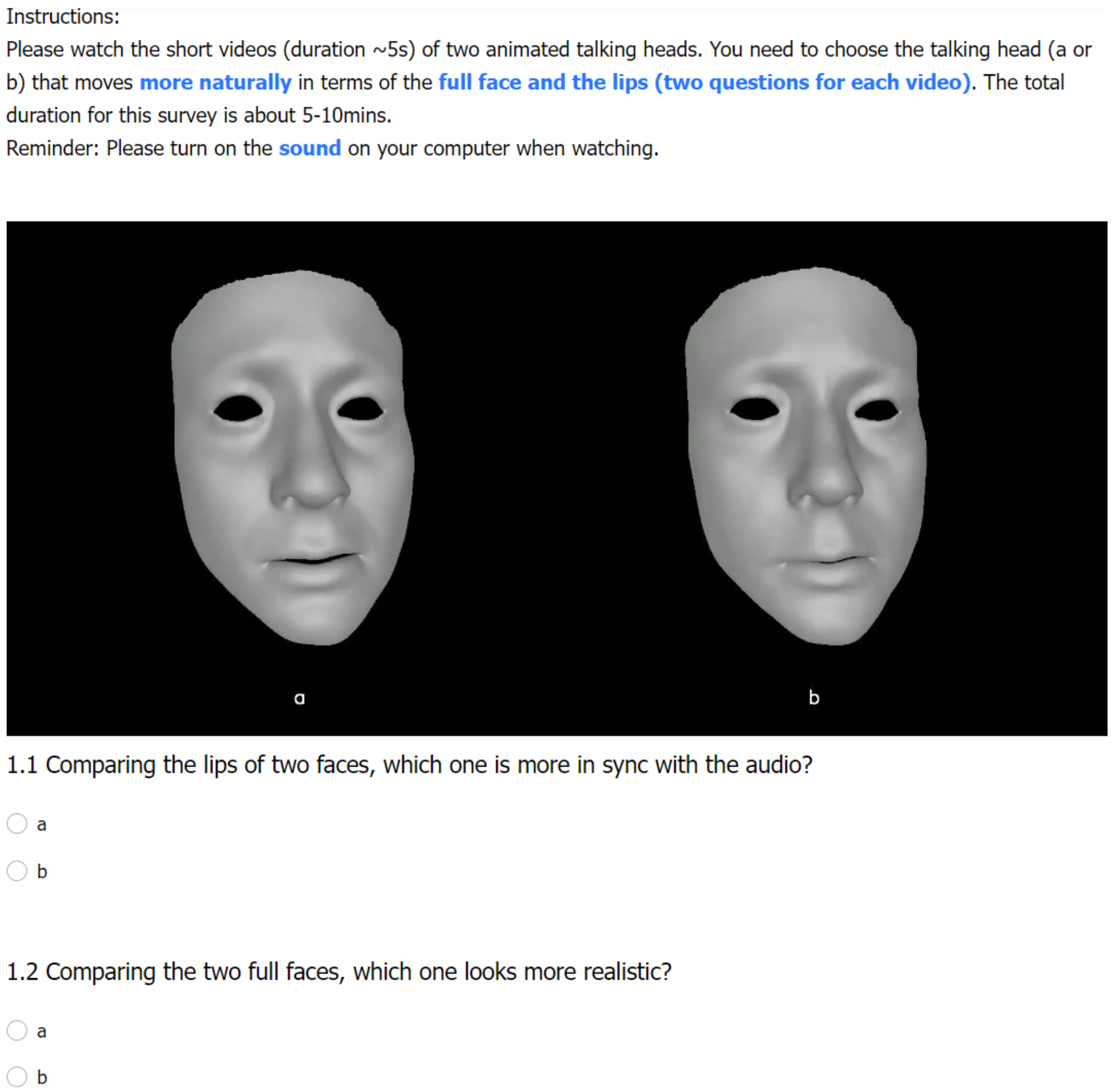
[1] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J Black. Capture, learning, and synthesis of 3d speaking styles. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[2] 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. 3

[3] 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. 1

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015. 1

* `https://doubiiu.github.io/projects/codetalker`.



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 3. 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.

[5] 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. 3

[6] 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

[7] 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. 3

[8] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 1

[9] 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. 1

[10] 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. 3