Imitator: Personalized Speech-driven 3D Facial Animation

Balamurugan Thambiraja[1] Ikhsanul Habibie[2] Sadegh Aliakbarian[3]
Darren Cosker[3] Christian Theobalt[2] Justus Thies[1]

[1] Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2] Max Planck Institute for Informatics, Saarland, Germany
[3] Mesh Labs, Microsoft, Cambridge, UK

In this supplemental document, we analyze the style adaptation with respect to the length of the reference video (see Sec. 1) and show an ablation study on 2-stage style-adaptation (Sec. 2), provide additional details of the proposed architecture (see Sec. 3), and discuss ethical considerations in Sec. 4.

1. Impact of Data to Style-Adaptation:

To analyze the impact of data on the style adaptation process, we randomly sample $(1, 4, 10, 20)$ sequences from the train set of the VOCA test subjects and perform our style adaption. Each sequence contains about $3-5$ seconds of data. In Tab. 1, we observe that the performance on the quantitative metrics increase with the number of reference sequences. As mentioned in the main paper, even an adaptation based on a single sequence results in a significantly better animation in comparison to the baseline methods. This highlights the impact of style on the generated animations.

Fig. 1 illustrates the lip distance curve for one test sequence used in this study. We observe that the lip distance with more reference data better fits the ground truth curve.

No. Seq.	Lip-Sync ↓	Lip-max ↓	L_2^{lip} ↓	L_2^{face} ↓
1	1.48	3.96	0.1	0.9
4	1.44	3.85	0.1	0.89
10	1.43	3.55	0.09	0.76
20	**1.35**	**3.43**	**0.09**	**0.69**

Table 1: Ablation of the style adaptation w.r.t. the amount of reference sequences used. With an increasing number of data, the quantitative metrics improve. Each sequence is $3-5$s long.

2. Ablation study on 2 stage Style-Adaptation:

Our proposed style adaptation has two stages as explained in the main paper Sec. 3.3. In the first step, we optimize for the style code and then we refine the motion basis and style code together. In Fig. 2, we show an example of the style adaptation by evaluating the lip distances throughout a sequence with a motion decoder at initialization, with optimized style code, and with a refined motion basis. While the lip distance with the generalized motion decoder is considerable, it gets significantly improved by the consecutive steps of style adaptation. After style code optimization, we observe that the amplitude and frequency of the lip distance curves start resembling the ground truth. From Tab. 2, we observe an increase in quantitative performance on *Lip-Sync* and *Lip-max* metrics. Refining the motion basis further improves the lip distance, and it is able to capture facial idiosyncrasies, like asymmetrical lip deformations. Quantitatively, it improves the metrics in the lip region significantly. However, as discussed in the main paper Sec. 5, we see a slight increase in the overall face error, when style-adaption is performed on fewer sequences ($\sim 20s$). This also gets improved when slightly more data ($\sim 50s$) is provided.

Method	Lip-Sync ↓	Lip-max ↓	L_2^{lip} ↓	L_2^{face} ↓
Initial Style	1.95	4.8	0.12	0.85
Style code optimization	1.81	4.53	0.12	**0.79**
Motion basis refinement	**1.44**	**3.85**	**0.1**	0.89

Table 2: Quantitative analysis of the different stages in our style-adaption pipeline. Note the ablation study is conducted on our proposed architecture and style-adaption is performed on 4 sequences.

3. Architecture Details

3.1. Audio Encoder:

Similar to Faceformer[3], our audio encoder is built upon the Wav2Vec 2.0 [1] architecture to extract temporal audio features. These audio features are fed into a linear interpo-

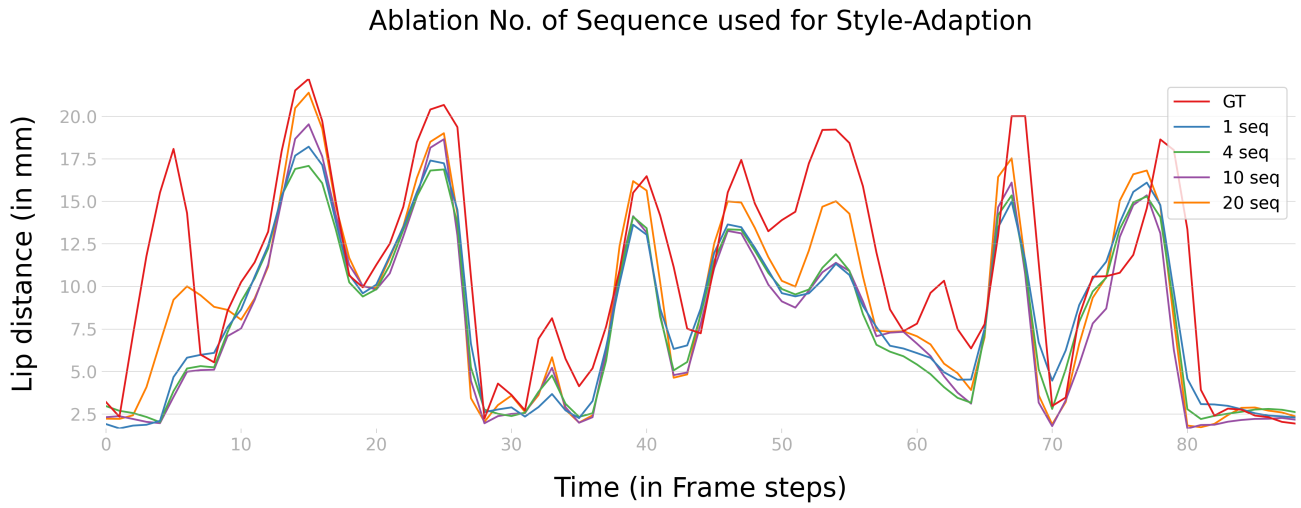

Figure 1: With an increasing number of reference data samples for style adaptation, the lip distance throughout a test sequence of VOCAset is approaching the ground truth lip distance curve.

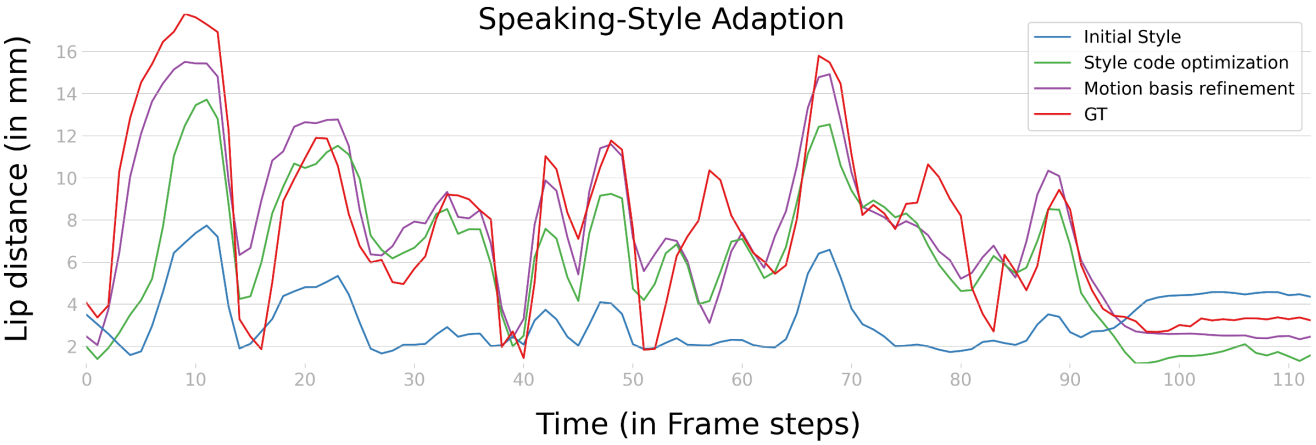

Figure 2: Analysis of style adaptation in terms of lip distance on a test sequence of the VOCAset [2] (reference in red). Starting from an initial talking style from the training set (blue), we consecutively adapt the style code (green) and the motion basis of the motion decoder (purple).

lation layer to convert the audio frequency to the motion frequency. The interpolated outputs are then fed into 12 identical transformer encoder layers with 12 attention heads and an output dimension of 768. A final linear projection layer converts the audio features from the 768-dimension features to a 64-dimensional phoneme representation.

3.2. Auto-regressive Viseme Decoder:

Our auto-regressive viseme decoder is built on top of traditional transformer decoder layers [5]. We use a zero vector of 64-dimension as a start token to indicate the start of sequence synthesis. We first add a positional encoding of 64-dimension to the input feature and fed it to decoder layers in the viseme decoder. For self-attention and cross-modal multi-head attention, we use 4 heads of dimension 64. Our feed forward layer dimension is 128.

Multi-Head Self-Attention: Given a sequence of positional encoded inputs $\hat{h}_t$, we use multi-head self-attention (self-MHA), which generates the context representation of the inputs by weighting the inputs based on their relevance. The Scaled Dot-Product attention function can be defined as mapping a query and a set of key-value pairs to an output, where queries, keys, values and outputs are vectors [5]. The output is the weighted sum of the values; the weight is computed by a compatibility function of a query with the

corresponding key. The attention can be formulated as:

$$Attention(Q, K, V) = \sigma(\frac{QK^T}{\sqrt{d_k}})V, \quad (1)$$

where Q, K, V are the learned Queries, Keys and Values, $\sigma(\cdot)$ denotes the softmax activation function, and d_k is the dimension of the keys. Instead of using a single attention mechanism and generating one context representation, MHA uses multiple self-attention heads to jointly generate multiple context representations and attend to the information in the different context representations at different positions. MHA is formulated as follows:

$$MHA(Q, K, V) = [head_1, ..., head_h] \cdot W^O, \quad (2)$$

with $head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$, where W^O, W_i^Q, W_i^K, W_i^V are weights related to each input variable.

Audio-Motion Multi-Head Attention The Audio-Motion Multi-Head attention aims to map the context representations from the audio encoder to the viseme representations by learning the alignment between the audio and style-agnostic viseme features. The decoder queries all the existing viseme features with the encoded audio features, which carry both the positional information and the contextual information, thus, resulting in audio context-injected viseme features. Similar to Faceformer [3], we add an alignment bias along the diagonal to the query-key attention score to add more weight to the current time audio features. The alignment bias $B^A(1 \leq i \leq t, 1 \leq j \leq KT)$ is:

$$B^A(i, j) = \begin{cases} 0 & \text{if } (i = j), \\ -\infty & \text{otherwise.} \end{cases} \quad (3)$$

The modified Audio-Motion Attention is represented as:

$$Attention(Q^v, K^a, V^a, B^A) = \sigma(\frac{Q^v(K^a)^T}{\sqrt{d_k}} + B^A)V^a, \quad (4)$$

where Q^v are the learned queries from viseme features, K^a the keys and V^a the values from the audio features, $\sigma(\cdot)$ is the softmax activation function, and d_k is the dimension of the keys.

3.3. Motion Decoder:

The motion decoder aims to generate 3D facial animations $\hat{y}_{1:T}$ from the style-agnostic viseme features $\hat{v}_{1:T}$ and a style embedding $\hat{S}_i$. Specifically, our motion decoder consists of two components, a style embedding layer and a motion synthesis block. The style linear layer takes a one-hot encoder of 8-dimension and produce a style-embedding of 64-dimension. The style-embedding is added to input viseme features and fed into 4 successive linear layers which have a leaky-ReLU as activation. The output dimension of the 4-layer block is 64 dimensional. A final fully connected layer maps the 64-dimension input features to the 3D face deformation described as per-vertex displacements of size 15069. This layer is defining the motion deformation basis of a subject and is adapted based on a reference sequence.

Training Details: We use the ADAM optimizer with a learning rate of 1e-4 for both the style-agnostic transformer training and the style adaptation stage. During the style-agnostic transformer training, the parameters of the Wave2Vec 2.0 layers in the audio encoder are fixed. Our model is trained for 300 epochs, and the best model is chosen based on the validation loss. During the style-adaptation stage, we first generate the viseme features and keep them fixed during the style adaptation stage. Then, we optimize for the style embedding for 300 epochs. Finally, the style-embedding and final motion deformation basis is refined for another 300 epochs. For generalized training, we use the following weights $\lambda_{MSE} = 1.0$, $\lambda_{vel} = 10.0$, and $\lambda_{lip} = 5.0$. For style-adaption on the VOCAset and external sequence, we use the $\lambda_{vel} = 1.0$ and $\lambda_{vel} = 10.0$ for best performance. Additionally, based on the speaking style of the target actor, we observed that training for longer epochs tends to improve expressiveness. However, for standard evaluation, we perform style-adaption for 300 epochs as explained earlier.

4. Broader Impact

Our proposed method aims at the synthesis of realistic-looking 3D facial animations. Ultimately, these animations can be used to drive photo-realistic digital doubles of people in audio-driven immersive telepresence applications in AR or VR. However, this technology can also be misused for so-called DeepFakes. Given a voice cloning approach, our method could generate 3D facial animations that drive an image synthesis method. This can lead to identity theft, cyber mobbing, or other harmful criminal acts. We believe that conducting research openly and transparently could raise awareness of the misuse of such technology. We will share our implementation to enable research on digital multi-media forensics. Specifically, synthesis methods are needed to produce the training data for forgery detection [4].

References

[1] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Infor-*

mation Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. 1

[2] Daniel Cudeiro, Timo Bolkart, Cassidy Laidlaw, Anurag Ranjan, and Michael J. Black. Capture, Learning, and Synthesis of 3D Speaking Styles. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10093–10103, Long Beach, CA, USA, June 2019. IEEE. 2

[3] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. Faceformer: Speech-driven 3d facial animation with transformers. *CoRR*, abs/2112.05329, 2021. 1, 3

[4] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. *ICCV 2019*, 2019. 3

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2