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] Microsoft Mixed Reality & AI Lab, Cambridge, UK

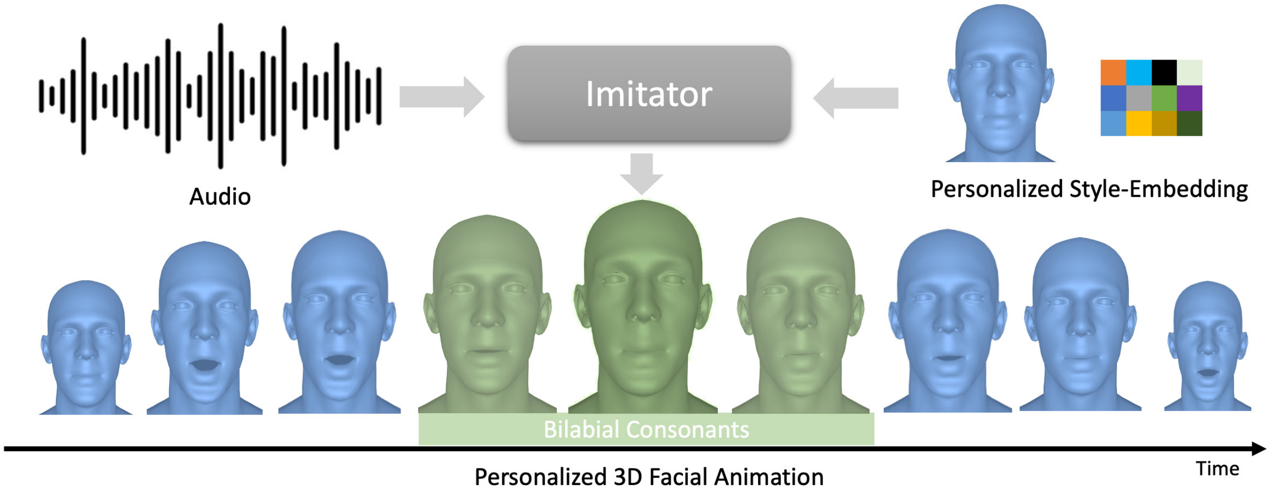
Imitator
Audio
Personalized Style-Embedding
Bilabial Consonants
Personalized 3D Facial Animation
Time

Figure 1. *Imitator* is a novel method for personalized speech-driven 3D facial animation. Given an audio sequence and a personalized style-embedding as input, we generate person-specific motion sequences with accurate lip closures for bilabial consonants ('m','b','p'). The style-embedding of a subject can be computed by a short reference video (e.g., 5s).

Abstract

Speech-driven 3D facial animation has been widely explored, with applications in gaming, character animation, virtual reality, and telepresence systems. State-of-the-art methods deform the face topology of the target actor to sync the input audio without considering the identity-specific speaking style and facial idiosyncrasies of the target actor, thus, resulting in unrealistic and inaccurate lip movements. To address this, we present Imitator, a speech-driven facial expression synthesis method, which learns identity-specific details from a short input video and produces novel facial expressions matching the identity-specific speaking style and facial idiosyncrasies of the target actor. Specifically, we train a style-agnostic transformer on a large facial expression dataset which we use as a prior for audio-driven facial expressions. Based on this prior, we optimize for identity-specific speaking style based on a short reference video. To train the prior, we introduce a novel loss function based on detected bilabial consonants to ensure plausible lip closures and consequently improve the realism of the generated expressions. Through detailed experiments and a user study, we show that our approach produces temporally coherent facial expressions from input audio while preserving the speaking style of the target actors. Please check out the project page for the supplemental video and more results.

1. Introduction

3D digital humans raised a lot of attention in the past few years as they aim to replicate the appearance and motion of real humans for immersive applications, like telepresence in AR or VR, character animation and creation for entertainment (movies and games), and virtual mirrors for e-commerce. Especially, with the introduction of neural rendering [27, 29], we see immense progress in the photo-realistic synthesis of such digital doubles [11,20,38]. These avatars can be controlled via visual tracking to mirror the facial expressions of a real human. However, we need to control the facial avatars with text or audio inputs for a series of applications. For example, AI-driven digital assistants rely on motion synthesis instead of motion cloning. Even telepresence applications might need to work with audio inputs only, when the face of the person is occluded or cannot be tracked, since a face capture device is not available. To this end, we analyze motion synthesis for facial animations from audio inputs; note that text-to-speech approaches can be used to generate such audio. Humans are generally sensitive towards faces, especially facial motions, as they are crucial for communication (e.g., micro-expressions). Without full expressiveness and proper lip closures, the generated animation will be perceived as unnatural and implausible. Especially if the person is known, the facial animations must match the subject's idiosyncrasies.

Recent methods for speech-driven 3D facial animation [5, 10, 16, 21] are data-driven. They are trained on high-quality motion capture data and leverage pretrained speech models [13,23] to extract an intermediate audio representation. We can classify these data-driven methods into two categories, generalized [5,10,21] and personalized animation generation methods [16]. In contrast to those approaches, we aim at a personalized 3D facial animation synthesis that can adapt to a new user while only relying on input RGB videos captured with commodity cameras. Specifically, we propose a transformer-based auto-regressive motion synthesis method that predicts a generalized motion representation. This intermediate representation is decoded by a motion decoder which is adaptable to new users. A speaker embedding is adjusted for a new user, and a new motion basis for the motion decoder is computed. Our method is trained on the VOCA dataset [5] and can be applied to new subjects captured in a short monocular RGB video. As lip closures are of paramount importance for bilabial consonants ('m','b','p'), we introduce a novel loss based on the detection of bilabials to ensure that the lips are closed properly. We take inspiration from the locomotion synthesis field [14,18], where similar losses are used to enforce foot contact with the ground and transfer it to our scenario of physically plausible lip motions.

In a series of experiments and ablation studies, we demonstrate that our method is able to synthesize facial expressions that match the target subject's motions in terms of style and expressiveness. Our method outperforms state-of-the-art methods in our metrical evaluation and user study. Please refer to our supplemental video for a detailed qualitative comparison. In a user study, we confirm that personalized facial expressions are important for the perceived realism.

The contributions of our work *Imitator* are as follows:

- a novel auto-regressive motion synthesis architecture that allows for adaption to new users by disentangling generalized viseme generation and person-specific motion decoding,
- and a lip contact loss formulation for improved lip closures based on physiological cues of bilabial consonants ('m','b','p').

2. Related Work

Our work focuses on speech-driven 3D facial animation related to talking head methods that create photo-realistic video sequences from audio inputs.

Talking Head Videos: Several prior works on speech-driven generation focus on the synthesis of 2D talking head videos. Suwajanakorn et al. [25] train an LSTM network on 19h video material of Obama to predict his person-specific 2D lip landmarks from speech inputs, which is then used for image generation. Vougioukas et al. [33] propose a method to generate facial animation from a single RGB image by leveraging a temporal generative adversarial network. Chung et al. [4] introduce a real-time approach to generate an RGB video of a talking face by directly mapping the audio input to the video output space. This method can redub a new target identity not seen during training. Instead of performing direct mapping, Zhou et al. [39] disentangles the speech information in terms of speaker identity and content, allowing speech-driven generation that can be applied to various types of realistic and hand-drawn head portraits. A series of work [24,28,36,37] uses an intermediate 3D Morphable Model (3DMM) [2,8] to guide the 2D neural rendering of talking heads from audio. Wang et al. [34] extend this work also to model the head movements of the speaker. Lipsync3d [17] proposes data-efficient learning of personalized talking heads focusing on pose and lighting normalization. Based on dynamic neural radiance fields [11], Ad-nerf [12] and DFA-NeRF [35] learn personalized talking head models that can be rendered under novel views, while being controlled by audio inputs. In contrast to these methods, our work focuses on predicting 3D facial animations from speech that can be used to drive 3D digital avatars without requiring retraining of the entire model to capture the person-specific motion style.

Speech-Driven 3D Facial Animation: Speech-driven 3d facial animation is a vivid field of research. Earlier methods [6, 7, 9, 15, 32] focus on animating a predefined facial rig using procedural rules. HMM-based models generate visemes from input text or audio, and the facial animations are generated using viseme-dependent co-articulation models [6, 7] or by blending facial templates [15]. With recent advances in machine learning, data-driven methods [3, 5, 10, 16, 21, 26, 28] have demonstrated their capability to learn viseme patterns from data. These methods

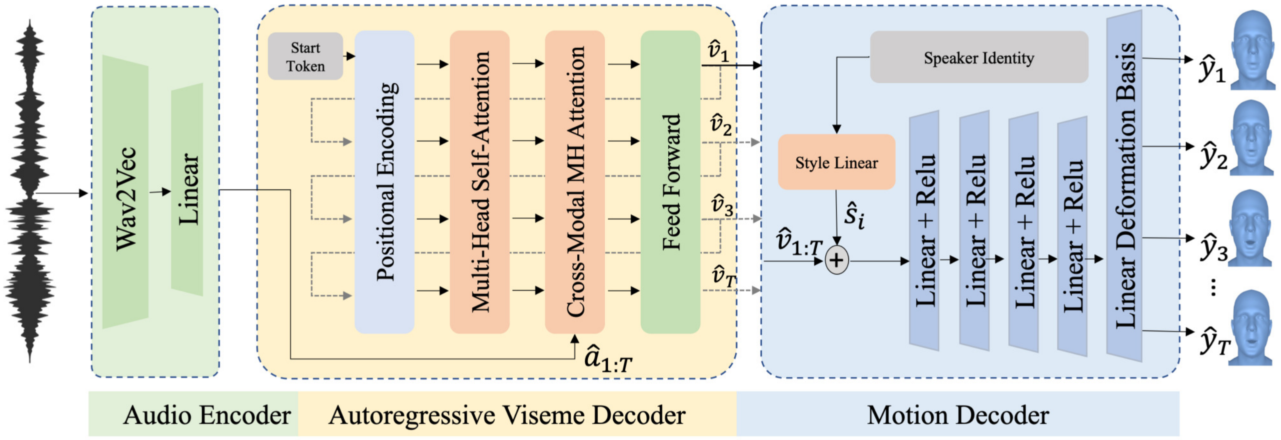

Figure 2. Our architecture takes audio as input which is encoded by a pre-trained Wav2Vec2.0 model [1]. This audio embedding $\hat{a}_{1:T}$ is interpreted by an auto-regressive viseme decoder which generates a generalized motion feature $\hat{v}_{1:T}$. A style-adaptable motion decoder maps these motion features to person-specific facial expressions $\hat{y}_{1:T}$ in terms of vertex displacements on top of a template mesh.

are based on pretrained speech models [1, 13, 23] to generate an abstract and generalized representation of the input audio, which is then interpreted by a CNN or auto-regressive model to map to either a 3DMM space or directly to 3D meshes. Karras et al. [16] learn a 3D facial animation model from 3-5 minutes of high-quality actor specific 3D data. VOCA [5] is trained on 3D data of multiple subjects and can animate the corresponding set of identities from input audio by providing a one-hot encoding during inference that indicates the subject. MeshTalk [21] is a generalized method that learns a categorical representation for facial expressions and auto-regressively samples from this categorical space to animate a given 3D facial template mesh of a subject from audio inputs. FaceFormer [10] uses a pretrained Wav2Vec [1] audio representation and applies a transformer-based decoder to regress displacements on top of a template mesh. Like VOCA, FaceFormer provides a speaker identification code to the decoder, allowing one to choose from the training set talking styles. In contrast, we aim at a method that can adapt to new users, capturing their talking style and expressiveness.

3. Method

Our goal is to model person-specific speaking style and the facial idiosyncrasies of an actor, to generate 3D facial animations of the subject from novel audio inputs. As input, we assume a short video sequence of the subject which we leverage to compute the identity-specific speaking style. To enable fast adaptation to novel users without significant training sequences, we learn a generalized style-agnostic transformer on VOCAset [5]. This transformer provides generic motion features from audio inputs that are interpretable by a person-specific motion decoder. The motion decoder is pre-trained and adaptable to new users via speaking style optimization and refinement of the motion basis. To further improve synthesis results, we introduce a novel lip contact loss based on physiological cues of the bilabial consonants [7]. In the following, we will detail our model architecture and the training objectives and describe the style adaptation.

3.1. Model Architecture

Our architecture consists of three main components (see Figure 2): an audio encoder, a generalized auto-regressive viseme decoder, and an adaptable motion decoder.

Audio Encoder: Following state-of-the-art motion synthesis models [5, 10], we use a generalized speech model to encode the audio inputs A. Specifically, we leverage the Wav2Vec 2.0 model [1]. The original Wav2Vec is based on a CNN architecture designed to produce a meaningful latent representation of human speech. To this end, the model is trained in a self-supervised and semi-supervised manner to predict the immediate future values of the current input speech by using a contrastive loss, allowing the model to learn from a large amount of unlabeled data. Wav2Vec 2.0 extends this idea by quantizing the latent representation and incorporating a Transformer-based architecture [31]. We re-sample the Wav2Vec 2.0 output with a linear interpolation layer to match the sampling frequency of the motion (30fps for the VOCAset, with 16kHz audio), resulting in a contextual representation $\{\hat{a}\}_{t=1}^{T}$ of the audio sequence for T motion frames.

Auto-regressive Viseme Decoder: The decoder F_v takes the contextual representation of the audio sequence as input and produces style agnostic viseme features $\hat{v}_t$ in an auto-regressive manner. These viseme features describe how the lip should deform given the context audio and the previous viseme features. In contrast to Faceformer [10], we propose to use of a classical transformer architecture [31] as viseme decoder, which learns the mapping from audio-features $\{\hat{a}\}_{t=1}^{T}$ to identity agnostic viseme features $\{\hat{v}\}_{t=1}^{T}$. The autoregressive viseme decoder is defined as:

$$\hat{v}_t = F_v(\theta_v; \hat{v}_{1:t-1}, \hat{a}_{1:T}), \tag{1}$$

where θ_v are the learnable parameters of the transformer.

In contrast to the traditional neural machine translation (NMT) architectures that produce discrete text, our output representation is a continuous vector. NMT models use a start and end token to indicate the beginning and end of the sequence. During inference, the NMT model autoregressively generates tokens until the end token is generated. Similarly, we use a start token to indicate the beginning of the sequences. However, since the sequence length T is given by the length of the audio input, we do not use an end token. We inject temporal information into the sequences by adding encoded time to the viseme feature in the sequence. We formulate the positionally encoded intermediate representations $\hat{h}_t$ as:

$$\hat{h}_t = \hat{v}_t + PE(t), \tag{2}$$

where $PE(t)$ is a sinusoidal encoding function [31]. Given the sequence of positional encoded inputs $\hat{h}_t$, we use multi-head self-attention which generates the context representation of the inputs by weighting the inputs based on their relevance. These context representations are used as input to a cross-modal multi-head attention block which also takes the audio features $\hat{a}_{1:T}$ from the audio encoder as input. A final feed-forward layer maps the output of this audio-motion attention layer to the viseme embedding $\hat{v}_t$. In contrast to Faceformer [10], which feeds encoded face motions $\hat{y}_t$ to the transformer, we work with identity-agnostic viseme features which are independently decoded by the motion decoder. We found that feeding face motions $\hat{y}_t$ via an input embedding layer to the transformer contains identity-specific information, which we try to avoid since we aim for a generalized viseme decoder that is disentangled from person-specific motion. In addition, using a general start token instead of the identity code [10] as the start token reduces the identity bias further. Note that disentangling the identity-specific information from the viseme decoder improves the motion optimization in the style adaption stage of the pipeline (see Section 3.3), as gradients do not need to be propagated through the auto-regressive transformer.

Motion Decoder: The motion decoder aims to generate 3D facial animation $\hat{y}_{1:T}$ 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. For the training of the style-agnostic transformer and for pre-training the motion decoder, we assume to have a one-hot encoding of the identities of the training set. The style embedding layer takes this identity information as input and produces the style embedding $\hat{S}_i$, which encodes the identity-specific motion. The style embedding is concatenated with the viseme features $\hat{v}_{1:T}$ and fed into the motion synthesis block. The motion synthesis block consists of non-linear layers which map the style-aware viseme features to the motion space defined by a linear deformation basis. During training, the deformation basis is learned across all identities in the dataset. The deformation basis is fine-tuned for style adaptation to out-of-training identities (see Section 3.3). The final mesh outputs $\hat{y}_{1:T}$ are computed by adding the estimated per-vertex deformation to the template mesh of the subject.

3.2. Training

Similar to Faceformer [10], we use an autoregressive training scheme instead of teacher-forcing to train our model on the VOCAset [5]. Given that VOCAset provides ground truth 3D facial animations, we define the following loss:

$$\mathcal{L}_{total} = \lambda_{MSE} \cdot \mathcal{L}_{MSE} + \lambda_{vel} \cdot \mathcal{L}_{vel} + \lambda_{lip} \cdot \mathcal{L}_{lip}, \tag{3}$$

where $\mathcal{L}_{MSE}$ defines a reconstruction loss of the vertices, $\mathcal{L}_{vel}$ defines a velocity loss, and $\mathcal{L}_{lip}$ measures lip contact. The weights are $\lambda_{MSE} = 1.0$, $\lambda_{vel} = 10.0$, and $\lambda_{lip} = 5.0$.

Reconstruction Loss: The reconstruction loss $\mathcal{L}_{MSE}$ is:

$$\mathcal{L}_{MSE} = \sum_{v=1}^{V} \sum_{t=1}^{T_v} ||y_{t,v} - \hat{y}_{t,v}||^2, \tag{4}$$

where $y_{t,v}$ is the ground truth mesh at time t in sequence v (of V total sequences) and $\hat{y}_{t,v}$ is the prediction.

Velocity Loss: Our motion decoder takes independent viseme features as input to produce facial expressions. To improve temporal consistency in the prediction, we introduce a velocity loss $\mathcal{L}_{vel}$ similar to [5]:

$$\mathcal{L}_{vel} = \sum_{v=1}^{V} \sum_{t=2}^{T_v} ||(y_{t,v} - y_{t-1,v}) - (\hat{y}_{t,v} - \hat{y}_{t-1,v})||^2. \tag{5}$$

Lip Contact Loss: Training with L_{MSE} guides the model to learn an averaged facial expression, thus resulting in improper lip closures. To this end, we introduce a novel lip contact loss for bilabial consonants ('m','b','p') to improve lip closures. Specifically, we automatically annotate the VOCAset to extract the occurrences of these consonants; see Section 4. Using this data, we define the following lip loss:

$$\mathcal{L}_{lip} = \sum_{t=1}^{T} \sum_{j=1}^{N} w_t ||y_{t,v} - \hat{y}_{t,v}||^2, \tag{6}$$

where $w_{t,v}$ weights the prediction of frame t according to the annotation of the bilabial consonants. Specifically, $w_{t,v}$ is one for frames with such consonants and zero otherwise.

Note that for such consonant frames, the target $y_{t,v}$ represents a face with a closed mouth; thus, this loss improves lip closures at 'm','b' and 'p's (see Section 5).

3.3. Style Adaptation

Given a video of a new subject, we reconstruct and track the face $\tilde{y}_{1:T}$ (see Section 4). Based on this reference data, we first optimize for the speaker style-embedding $\hat{S}$ and then jointly refine the linear deformation basis using the $\mathcal{L}_{MSE}$ and $\mathcal{L}_{vel}$ loss. In our experiments, we found that this two-stage adaptation is essential for generalization to new audio inputs as it reuses the pretrained information of the motion decoder. As an initialization of the style embedding, we use a speaking style of the training set. We precompute all viseme features $\hat{v}_{1:T}$ once, and optimize the speaking style to reproduce the tracked faces $\tilde{y}_{1:T}$. We then refine the linear motion basis of the decoder to match the person-specific deformations (e.g., asymmetric lip motions).

4. Dataset

We train our method based on the VOCAset [5], which consists of 12 actors (6 female and 6 male) with 40 sequences each with a length of $3-5$ seconds. The dataset comes with a train/test set split which we use in our experiments. The test set contains 2 actors. The dataset offers audio and high-quality 3D face reconstructions per frame (60fps). For our experiment, we sample the 3D face reconstructions at 30fps. We train the auto-regressive transformer on this data using the loss from Equation (3). For the lip contact loss L_{lip}, we automatically compute the labels as described below.

To adapt the motion decoder to a new subject, we require a short video clip of the person. Using this sequence, we run a 3DMM-based face tracker to get the per-frame 3D shape of the person. Based on this data, we adapt the motion decoder as detailed in Section 3.3.

Automatic Lip Closure Labeling: For the VOCAset, the transcript is available. Based on Wav2Vec features, we align the transcript with the audio track. As the lip closure is formed before we hear the bilabial consonants, we search for the lip closure in the tracked face geometry before the time-stamp of the occurrence of the consonants in the script. We show this process for a single sequence in Figure 3. The lip closure is detected by lip distance, i.e., the frame with minimal lip distance in a short time window before the consonant is assumed to be the lip closure.

External Sequence Processing: We assume to have a monocular RGB video of about 2 minutes in length as input which we divide into train/validation/test sequences. Based on MICA [40], we estimate the 3D shape of the subject using the first frame of the video. Using this shape estimate, we run an analysis-by-synthesis approach [30] to estimate per-frame blendshape parameters of the FLAME 3DMM [19]. Given these blendshape coefficients, we can compute the 3D vertices of the per-frame face meshes that we need to adapt the motion decoder. Note that in contrast to the training data of the transformer, we do not require any bilabial consonants labeling, as we adapt the motion decoder only based on the reconstruction and velocity loss.

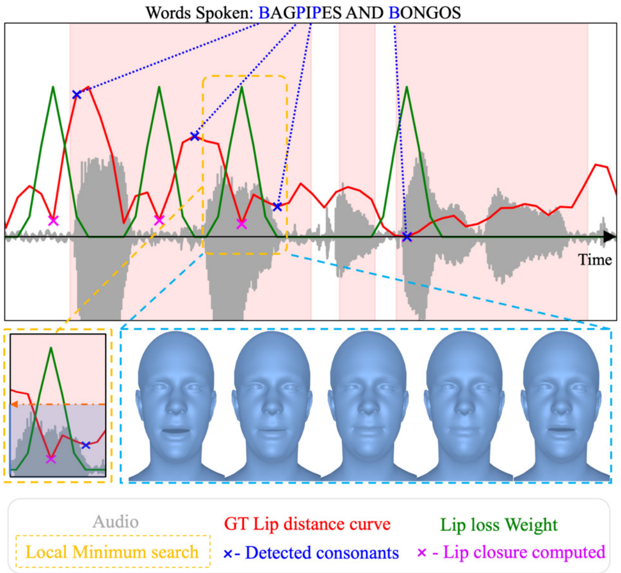

Figure 3. Automatic labeling of the bilabial consonants ('m','b' and 'p') and their corresponding lip closures in a sequence of VOCAset [5]. We align the transcript with the audio track using Wav2vec [1] features and extract the time stamps for the bilabial consonants. To detect the lip closures for the bilabial consonants, we search for local-minima on the Lip distance curves (red). The lip loss weights $w_{t,v}$ in a window around the detected lip closure are set to fixed values of a Gaussian function. We show an example of detected lip closures in the figure (in the blue bounding box).

5. Results

To validate our method, we conducted a series of qualitative and quantitative evaluations, including a user study and ablation studies. For evaluation on the test set of VOCAset [5], we randomly sample 4 sequences from the test subjects' train set (each $\sim$ 5s long) and learn the speaking-style and facial idiosyncrasies of the subject via style adaptation. We compare our method to the state-of-the-art methods VOCA [5], Faceformer [10], and MeshTalk [21]. We use the original implementations of the authors. However, we found that MeshTalk cannot train on the comparably small VOCAset. Thus, we qualitatively compare against MeshTalk with their provided model trained on a large-scale proprietary dataset with 200 subjects and 40 sequences for each. Note that the pretrained MeshTalk model is not compatible with the FLAME topology; thus, we cannot evaluate their method on novel identities. In addition to the experi-

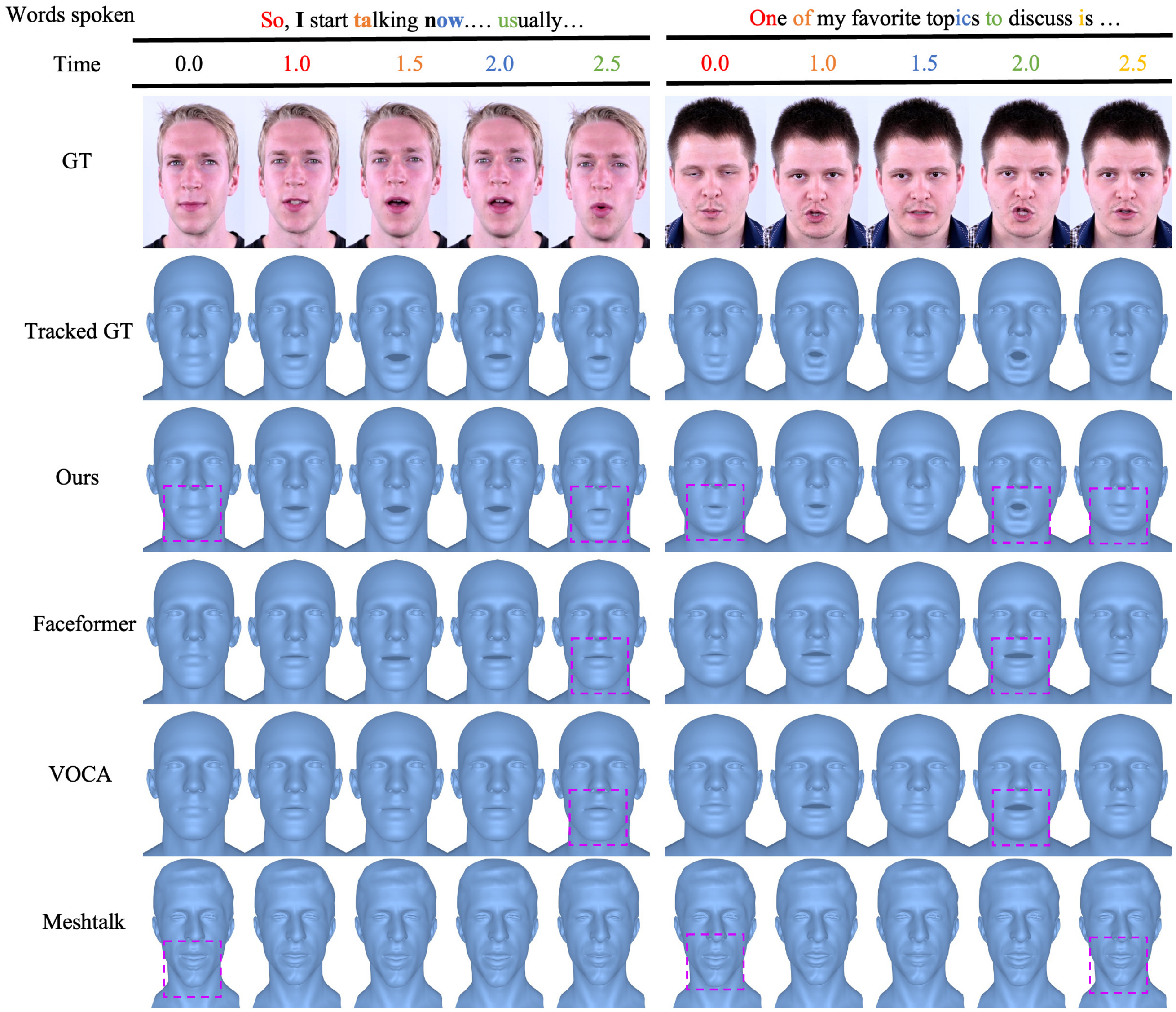

Figure 4. Qualitative comparison to the state-of-the-art methods VOCA [5], Faceformer [10], and MeshTalk [21]. Note that *MeshTalk is performed with a different identity* since we use their pretrained model, which cannot be trained on VOCAset. As we see in the highlighted regions, the geometry of the generated sequences without the person-specific style have muted and inaccurate lip animations.

ments on the VOCAset, we show results on external RGB sequences. The results can be best seen in the suppl. video.

Quantitative Evaluation: To quantitatively evaluate our method, we use the test set of VOCAset [5], which provides high-quality reference mesh reconstructions. We evaluate the performance of our method based on a mean L_2 vertex distance for the entire mesh L_2^{face} and the lip region L_2^{lip}. Following MeshTalk [21], we also compute the Lip-sync, which measures the mean of the maximal per-frame lip distances. In addition, we use Dynamic Time Wrapping (DTW) to compute the similarity between the produced and reference meshes, both for the entire mesh (F-DTW) and the lip region (Lip-DTW). Since VOCA and Faceformer do not adapt to new user talking styles, we select the talking style from their training with the best quantitative metrics. Note that the pretrained MeshTalk model is not applicable to this

Method	L_2^{face} ↓	L_2^{lip} ↓	**F-DTW** ↓	**Lip-DTW** ↓	**Lip-sync** ↓
VOCA [5]	0.88	0.15	1.28	2.41	5.72
Faceformer [10]	**0.8**	0.14	**1.18**	2.85	5.41
Ours (w/ 1seq)	0.91	0.1	1.3	1.68	3.99
Ours	0.89	**0.09**	1.26	**1.47**	**3.78**

Table 1. Quantitative results on the VOCAset [5]. Our method outperforms the baselines on all of the lip metrics while performing on par on the full-face metrics. Note that we are not targeting the animation of the upper face but aim for expressive and accurate lip movements, which is noticeable from the improved lip scores.

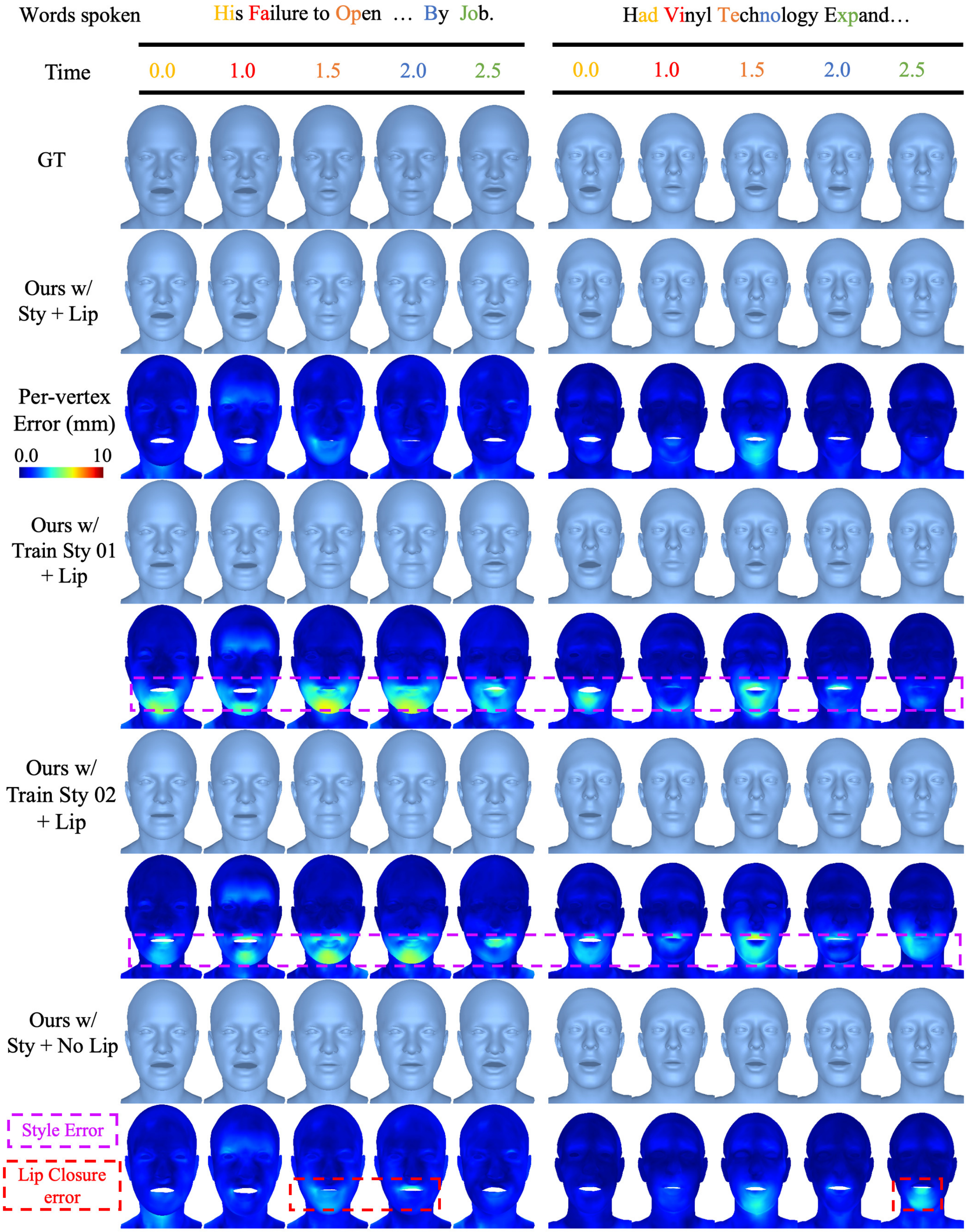

Figure 5. Qualitative ablation comparison. At first, we show that our complete method with style and $\mathcal{L}_{lip}$ loss is able to generate personalized facial animation with expressive motion and accurate lip closures. Replacing the person-specific style with the style seen during training results in generic and muted facial animation. As highlighted in the per-vertex error maps (magenta), the generated expression is not similar to the target actor. Especially the facial deformations are missing person-specific details. Removing $\mathcal{L}_{lip}$ from the training objective results in improper lip closures (red).

Method	**Expressiveness** (%)	**Realism/Lip-sync** (%)
Ours vs VOCA [5]	86.48	76.92
Ours vs Faceformer [10]	81.89	75.46
Ours vs Ground truth	20.28	42.30

Table 2. In a perceptual A/B user study conducted on the test set of VOCAset [5] with 56 participants, we see that in comparison to VOCA [5] and Faceformer [10] our method is preferred.

evaluation due to the identity mismatch. As can be seen in Table 1, our method achieves the lowest lip reconstruction and lip-sync errors, confirming our qualitative results. Even when using a single reference video for style adaptation (5s), our results shows significantly better lip scores.

Qualitative Evaluation: We conducted a qualitative evaluation on external sequences not part of VOCAset. In Figure 4, we show a series of frames from those sequences with the corresponding words. As we can see, our method is able to adapt to the speaking style of the respective subject. VOCA [5] and Faceformer [10] miss person-specific deformations and are not as expressive as our results. MeshTalk [21], which uses an identity that comes with the pretrained model, also shows dampened expressivity. In the suppl. video, we can observe that our method is generating better lip closures for bilabial consonants.

Perceptual Evaluation: We conducted a perceptual evaluation to quantify the quality of our method's generated results (see Table 2). Specifically, we conducted an A/B user study on the test set of VOCAset. We randomly sample 10 sequences of the test subjects and run our method, VOCA, and Faceformer. For VOCA and Faceformer, which do not adapt to the style of a new user, we use the talking style of the training Subject 137, which provided the best quantitative results. We use 20 videos per method resulting in 60 A/B comparisons. For every A/B test, we ask the user to choose the best method based on realism and expressiveness, following the user study protocol of Faceformer [10]. In Table 2, we show the result of this study in which 56 people participated. We observe that our method consistently outperforms VOCA and Faceformer. We also see that our model achieves similar realism and lip-sync as ground truth. Note that the users in the perceptual study have not seen the original talking style of the actors before. However, the results show that our personalized synthesis leads to more realistic-looking animations.

5.1. Ablation Studies

To understand the impact of our style adaptation and the novel lip contact loss $\mathcal{L}_{lip}$ on the perceptual quality, we show a qualitative ablation study including per-vertex error maps in Figure 5. As highlighted in the figure, the style adaptation is critical to match the person-specific deformations and mouth shapes and improves expressiveness.

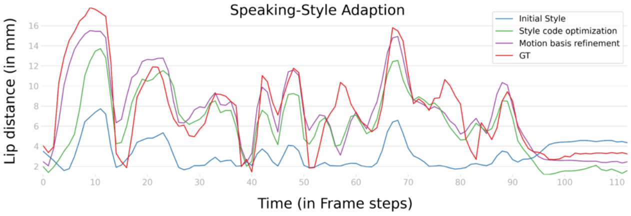

Figure 6. Analysis of style adaptation in terms of lip distance on a test sequence of the VOCAset [5] (reference in red). Starting from an initial talking style from the training set (blue), we consecutively adapt the style code (green) and the motion basis of the motion decoder (purple).

The lip contact loss improves the lip closures for the bilabial consonants, thus, improving the perceived realism, as can best be seen in the suppl. video. We rely on only $\sim$ 60 seconds-long reference videos to extract the person-specific speaking style. A detailed analysis of the sequence length's influence on the final output quality can be found in the suppl. material. It is also worth noting that our style-agnostic architecture allows us to perform style adaptation of the motion decoder in less than 30min, while an adaptation with an identity-dependent transformer takes about 6h.

Our proposed style adaptation has two stages as explained in Section 3.3. In the first step, we optimize for the style code and the refine the motion basis. In Figure 6, 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. Refining the motion basis further improves the lip distance, and it is able to capture facial idiosyncrasies, like asymmetrical lip deformations.

6. Discussion

Our evaluation shows that our proposed method outperforms state-of-the-art methods in perceived expressiveness and realism. However, several limitations remain. Specifically, we only support the speaking style of the subject seen in the reference video and do not control the talking style w.r.t. emotions (e.g., sad, happy, angry). The viseme transformer and the motion decoder could be conditioned on an emotion flag; we leave this for future work. The expressiveness and facial details depend on the face tracker's quality; if the face tracking is improved, our method will predict better face shapes.

7. Conclusion

We present *Imitator*, a novel approach for personalized speech-driven 3D facial animation. Based on a short reference video clip of a subject, we learn a personalized motion decoder driven by a generalized auto-regressive transformer that maps audio to intermediate viseme features. Our studies show that personalized facial animations are essential for the perceived realism of a generated sequence. Our new loss formulation for accurate lip closures of bilabial consonants further improves the results. We believe that personalized facial animations are a stepping stone towards audio-driven digital doubles.

8. Acknowledgements

This project has received funding from the Mesh Labs, Microsoft, Cambridge, UK. Further, we would like to thank Berna Kabadayi, Jalees Nehvi, Malte Prinzler and Wojciech Zielonka for their support and valuable feedback. The authors thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Balamurugan Thambiraja.

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 Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3, 5, 11

[2] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999. 2

[3] Yong Cao, Wen C. Tien, Petros Faloutsos, and Frédéric Pighin. Expressive speech-driven facial animation. *ACM Trans. Graph.*, 24(4):1283–1302, oct 2005. 2

[4] Joon Son Chung, Amir Jamaludin, and Andrew Zisserman. You said that? *arXiv preprint arXiv:1705.02966*, 2017. 2

[5] 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, 4, 5, 6, 8

[6] José Mario De Martino, Léo Pini Magalhães, and Fábio Violaro. Facial animation based on context-dependent visemes. *Computers & Graphics*, 30(6):971–980, Dec. 2006. 2

[7] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: an animator-centric viseme model for expressive lip synchronization. *ACM Trans. Graph.*, 35:127:1–127:11, 2016. 2, 3

[8] Bernhard Egger, William AP Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, et al. 3d morphable face models—past, present, and future. *ACM Transactions on Graphics (TOG)*, 39(5):1–38, 2020. 2

[9] T. Ezzat and T. Poggio. MikeTalk: a talking facial display based on morphing visemes. In *Proceedings Computer Animation '98 (Cat. No.98EX169)*, pages 96–102, Philadelphia, PA, USA, 1998. IEEE Comput. Soc. 2

[10] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. Faceformer: Speech-driven 3d facial animation with transformers. *CoRR*, abs/2112.05329, 2021. 2, 3, 4, 5, 6, 8, 11, 12

[11] Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. *CoRR*, abs/2012.03065, 2020. 2

[12] Yudong Guo, Keyu Chen, Sen Liang, Yongjin 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

[13] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng. DeepSpeech: Scaling up end-to-end speech recognition. 12 2014. 2, 3

[14] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics (TOG)*, 35(4):1–11, 2016. 2

[15] G.A. Kalberer and L. Van Gool. Face animation based on observed 3D speech dynamics. In *Proceedings Computer Animation 2001. Fourteenth Conference on Computer Animation (Cat. No.01TH8596)*, pages 20–251, Seoul, South Korea, 2001. IEEE Comput. Soc. 2

[16] 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*, 36(4):1–12, July 2017. 2, 3

[17] 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, 2021. 2

[18] Jehee Lee, Jinxiang Chai, Paul SA Reitsma, Jessica K Hodgins, and Nancy S Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pages 491–500, 2002. 2

[19] 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, (Proc. SIGGRAPH Asia)*, 36(6), 2017. 5

[20] Stephen Lombardi, Tomas Simon, Jason Saragih, Gabriel Schwartz, Andreas Lehrmann, and Yaser Sheikh. Neural volumes: Learning dynamic renderable volumes from images. *ACM Trans. Graph.*, 38(4):65:1–65:14, July 2019. 2

[21] Alexander Richard, Michael Zollhofer, Yandong Wen, Fernando de la Torre, and Yaser Sheikh. MeshTalk: 3D Face Animation from Speech using Cross-Modality Disentanglement. In *2021 IEEE/CVF International Conference on*

Computer Vision (ICCV), pages 1153–1162, Montreal, QC, Canada, Oct. 2021. IEEE. 2, 3, 5, 6, 8

[22] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner. Faceforensics++: Learning to detect manipulated facial images. *ICCV 2019*, 2019. 12

[23] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. wav2vec: Unsupervised pre-training for speech recognition. In Gernot Kubin and Zdravko Kacic, editors, *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pages 3465–3469. ISCA, 2019. 2, 3

[24] Linsen Song, Wayne Wu, Chen Qian, Ran He, and Chen Change Loy. Everybody's talkin': Let me talk as you want. *IEEE Transactions on Information Forensics and Security*, 17:585–598, 2022. 2

[25] Supasorn Suwajanakorn, Steven M Seitz, and Ira Kemelmacher-Shlizerman. Synthesizing obama: learning lip sync from audio. *ACM Transactions on Graphics (ToG)*, 36(4):1–13, 2017. 2

[26] Sarah L. Taylor, Taehwan Kim, Yisong Yue, Moshe Mahler, James Krahe, Anastasio Garcia Rodriguez, Jessica K. Hodgins, and Iain A. Matthews. A deep learning approach for generalized speech animation. *ACM Trans. Graph.*, 36(4):93:1–93:11, 2017. 2

[27] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Yifan Wang, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, Tomas Simon, Christian Theobalt, Matthias Niessner, Jonathan T. Barron, Gordon Wetzstein, Michael Zollhoefer, and Vladislav Golyanik. Advances in neural rendering. 2022. 1

[28] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. Neural voice puppetry: Audio-driven facial reenactment. *ECCV 2020*, 2020. 2

[29] J. Thies, A. Tewari, O. Fried, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. State of the art on neural rendering. *EG*, 2020. 1

[30] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. 2020. 5

[31] 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. 3, 4, 11

[32] A. Verma, N. Rajput, and L.V. Subramaniam. Using viseme based acoustic models for speech driven lip synthesis. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, volume 5, pages V–720–3, Hong Kong, China, 2003. IEEE. 2

[33] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. Realistic speech-driven facial animation with gans. *International Journal of Computer Vision*, 128(5):1398–1413, 2020. 2

[34] S Wang, L Li, Y Ding, C Fan, and X Yu. Audio2head: Audio-driven one-shot talking-head generation with natural head motion. In *International Joint Conference on Artificial Intelligence*. IJCAI, 2021. 2

[35] Shunyu Yao, RuiZhe Zhong, Yichao Yan, Guangtao Zhai, and Xiaokang Yang. Dfa-nerf: Personalized talking head generation via disentangled face attributes neural rendering. *arXiv preprint arXiv:2201.00791*, 2022. 2

[36] 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

[37] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. Flow-guided one-shot talking face generation with a high-resolution audio-visual dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3661–3670, 2021. 2

[38] Yufeng Zheng, Victoria Fernández Abrevaya, Xu Chen, Marcel C. Bühler, Michael J. Black, and Otmar Hilliges. I M avatar: Implicit morphable head avatars from videos. *CoRR*, abs/2112.07471, 2021. 2

[39] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis, and Dingzeyu Li. Makelttalk: speaker-aware talking-head animation. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020. 2

[40] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces. ECCV, 2022. 5

Imitator: Personalized Speech-driven 3D Facial Animation – Supplemental Document –

9. 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 Table 3, 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.

Figure 7 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.	$L_2^{face} \downarrow$	$L_2^{lip} \downarrow$	**F-DTW** $\downarrow$	**Lip-DTW** $\downarrow$	**Lip-sync** $\downarrow$
1	0.91	0.1	1.3	1.68	3.99
4	0.89	0.1	1.26	1.47	3.78
10	0.76	0.09	1.07	1.37	3.57
20	0.7	0.09	0.99	1.27	3.49

Table 3. 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.

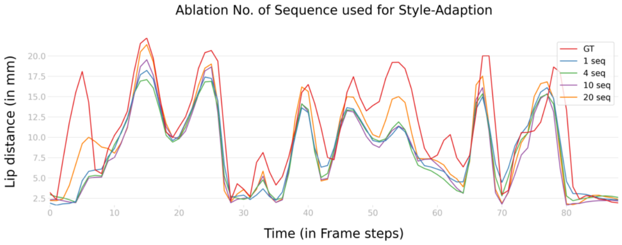

Figure 7. With an increasing number of reference data samples for style adaptation, the lip distance throughout a test sequence of VOCAset is approaching the ground truth lip distance curve.

10. Architecture Details

10.1. Audio Encoder:

Similar to Faceformer [10], 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 interpolation 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.

10.2. Auto-regressive Viseme Decoder:

Our auto-regressive viseme decoder is built on top of traditional transformer decoder layers [31]. 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 [31]. 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:

$$Attention(Q, K, V) = \sigma(\frac{QK^T}{\sqrt{d_k}})V, \tag{7}$$

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:

$$MHA(Q, K, V) = [head_1, ..., head_h] \cdot W^O, \tag{8}$$

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 [10], 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:

$$B^A(i,j) = \begin{cases} 0 & \text{if } (i = j), \\ -\infty & \text{otherwise.} \end{cases} \tag{9}$$

The modified Audio-Motion Attention is represented as:

$$Attention(Q^v, K^a, V^a, B^A) = \sigma(\frac{Q^v (K^a)^T}{\sqrt{d_k}} + B^A) V^a, \tag{10}$$

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.

10.3. 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 input viseme features are concatenated with the style-embedding 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 reconstruction 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.

11. 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 the 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 [22].

All participants in the study have given written consent to the usage of their video material for this publication.