MemoryTalker: Personalized Speech-Driven 3D Facial Animation via Audio-Guided Stylization

Supplementary Material

A. Overview

This supplementary document provides additional experiments and explanations to complement the main manuscript. We present a comprehensive performance comparison focusing on the effects of different memory slot sizes in our model. Additionally, we offer quantitative results comparing end-to-end training versus our proposed 2-stage training strategy. The document also includes extensive qualitative results, featuring detailed frame-by-frame comparisons between our method and existing approaches for different speakers under various conditions. Furthermore, we provide an in-depth feature analysis, visualizing how our method effectively captures speaking styles from audio input effectively. Lastly, we detail the implementation specifics of our MemoryTalker architecture and describe the methodology of our user study for a thorough evaluation against other methods.

B. Quantitative Results

B.1. Effects of Memory Slot Size

We conduct experiments varying the number of memory slots. Figure S1 shows Lip Vertex Errors (LVE) across different slot size configurations. We validate the performance while varying the memory slot size to (16, 24, 32, 48, 64). The optimal performance is achieved with 32 slots (indicated by star marker in Figure S1). Note that we leverage memory slot size 32 for our experiments in the main manuscript.

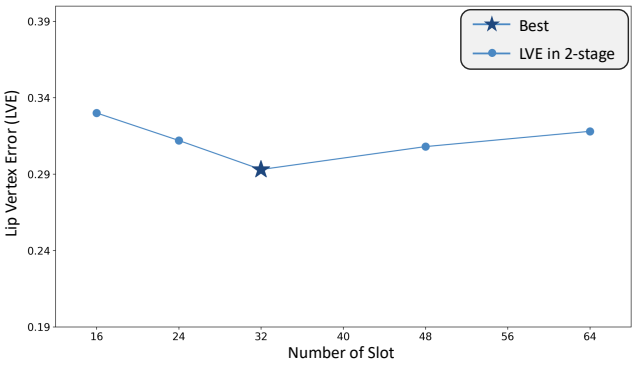

Figure S1. Effects of the memory slot size on LVE for facial animation generation.

Training Strategy	FVE ↓ ($\times 10^{-6}$)	LVE ↓ ($\times 10^{-5}$)	FID ↓ ($\times 10^{-1}$)
End-to-End	0.510	0.303	3.142
Two-Stage (Ours)	**0.506**	**0.293**	**3.045**

Table S1. Performance comparison between end-to-end learning and two-stage training strategy.

Method	FVE ↓ ($\times 10^{-6}$)	LVE ↓ ($\times 10^{-5}$)	LDTW ↓ ($\times 10^{-5}$)	Lip-max ↓ ($\times 10^{-4}$)
Mesh-driven [44]	0.673	0.400	0.521	0.431
Audio-driven (Ours)	**0.506**	**0.293**	**0.418**	**0.331**

Table S2. Quantitative results about mesh-based method.

B.2. End-to-End Learning vs. Two-stage Training Strategy

We propose a novel two-stage training strategy that first induces general motion synthesis, followed by speaking style adaptation. Table S1 shows performance comparisons between end-to-end learning and our two-stage training strategy. As seen in the Table S1, our training strategy consistently outperforms the end-to-end learning across all evaluation metrics, including FVE, LVE, and FID. These results validate that our strategy more effectively captures and reproduces personalized speaking styles while maintaining motion accuracy.

B.3. Comparison with Mesh-based Methods

As presented in Table S2, we conduct a quantitative comparison with an additional mesh-based method [44]. The results clearly indicate that our audio-driven approach achieves superior performance across all evaluation metrics. It is worth noting that Mimic [19], another mesh-based method, could not be directly compared as it utilizes different datasets for training and evaluation.

B.4. Evaluation on Seen Identities

We further evaluate our model's performance on seen identities, with the results shown in Table S3. The evaluation is performed on the BIWI Test-A set. We compare our **MemoryTalker** with CodeTalker [45], a one-hot-based method, and UniTalker [11], the current state-of-the-art approach. For a fair comparison, the correct one-hot identity vector is

Method	FVE↓ ($\times 10^{-1}$)	LVE↓ ($\times 10^{-3}$)	LDTW↓ ($\times 10^{-5}$)	Lip-max↓ ($\times 10^{-1}$)
CodeTalker [45]	0.122	0.453	0.637	0.356
UniTalker [11]	0.101	0.283	0.569	0.305
MemoryTalker (Ours)	**0.094**	**0.266**	**0.557**	**0.285**

Table S3. Quantitative results about seen identities.

Method	FVE ↓ ($\times 10^{-6}$)	LVE ↓ ($\times 10^{-5}$)	LDTW↓ ($\times 10^{-5}$)	Params (M)
ASR for E_s	0.518	0.300	0.437	185
MemoryTalker (Ours)	**0.506**	**0.293**	**0.418**	**94**

Table S4. Comparison the style features extracted by Mel-spectrogram- and ASR.

provided to CodeTalker. As demonstrated in the table, our method outperforms both competing methods in the seen identity setting as well.

B.5. Analysis on Speaking Style Features

We justify our choice of using mel-spectrograms for the speaking style encoder. Mel-spectrograms are well-known for preserving rich acoustic details such as prosody, rhythm, and timbre [38]. In contrast, features from an Automatic Speech Recognition (ASR) model are less suitable for this task because ASR models are trained to extract neutral text representations, thereby suppressing speaker-specific identity and style.

To empirically validate this, we present an ablation study in Table S4, where we replace our mel-spectrogram-based style encoder with an ASR-based one. The results show a degradation in performance, confirming that mel-spectrograms are better suited for capturing the nuances of speaking style for our task.

C. Qualitative Results

C.1. Performances Comparisons with Existing Methods

Figures S2 and S3 show extensive qualitative results, including detailed performance comparisons with existing methods. Our analysis provides a frame-by-frame visualization, offering a more nuanced and comprehensive comparison between our proposed method and current state-of-the-art approaches. Figure S2 illustrates cases where the pronunciation necessitates the opening of the mouth, exemplified by sounds like /a/. Conversely, Figure S3 showcases instances where the pronunciation involves the initial stages of lip closure, as demonstrated by sounds such as /o/. Upon careful examination of the error maps, it becomes evident that our proposed method achieves significantly more accurate results across both pronunciation types and diverse speaker profiles. This superior performance is consistently maintained when compared to all existing methods in the field. These results indicate that our method effectively captures and generates the respective speaking styles of multiple people under various different conditions.

C.2. The Effectiveness of Stylized Motion Memory in 2-Stage

Figure S7 shows the effectiveness of the stylized motion memory in 2-stage. In Figure S7, the first row shows the facial mesh and error map rendered by the *general* motion feature retrieved from motion memory $\mathbf{M}_m$ in 1-stage. On the other hand, the fourth row shows the facial mesh and error map rendered by the *personalized* motion feature recalled from the stylized motion memory $\tilde{\mathbf{M}}_m$ in 2-stage. The second and third rows show the magnified versions of the lip regions of the first and fourth rows, respectively. As shown in Figure S7, the limited lip movement can be observed in the first and second rows (results of 1-stage only, i.e., learning without 2-stage) compared to the third and fourth rows (results of learning with 2-stage). These results show that the motion memory $\mathbf{M}_m$ in 1-stage has a limitation in representing the subtle personal speaking style. On the other hand, by recalling the personalized motion feature from the stylized motion memory $\tilde{\mathbf{M}}_m$ in 2-stage, the 3D facial mesh can achieve fewer errors reflecting the individual's delicate speaking style.

C.3. Applying Style feature to General Motion

Figure S4 demonstrates the effectiveness of reflecting the style feature in 2-stage. In Figure S4, the first row shows the generated 3D facial motion from synchronized with audio at 1-stage. At this time, general motion corresponding to the audio is synthesized. On the other hand, the second, thrid, fourth, and fifth row demonstrates when the style features generated differently for each person in the 2-stage were applied to the general motion generated in the 1-stage, the subtle changed lip shape in the 2-stage was visualized. Through this, we verify that our method not only supplements the information lacking in general motion through speaking style features, but also effectively learns speaking style information for each speaker.

To demonstrate that our model performs audio-driven stylization rather than merely mirroring a fixed template, we applied different audio clips to the same identity. As shown in Figure S5, the generated animations not only produce lip motions that are accurately synchronized with each speech input but also reflect speaker-specific styles embedded in the audio—such as variations in pronunciation strength and articulation—while consistently preserving the target pro-

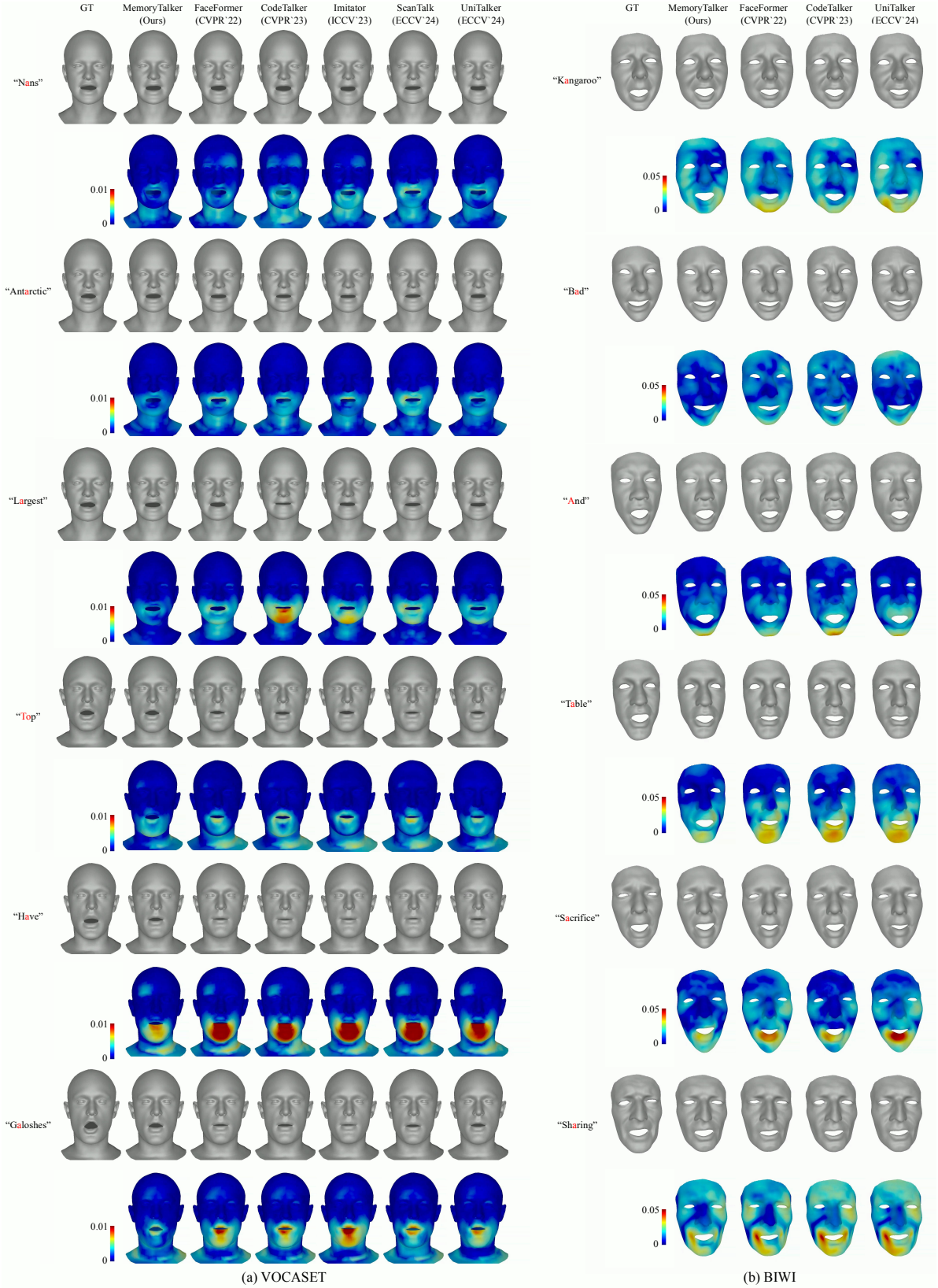

Figure S2. Qualitative comparisons for the cases where the pronunciation involves starting to OPENING the mouth.

Figure S3. Qualitative comparisons for the cases where the pronunciation involves starting to CLOSING the lips.

Module	Input→Output	Operation
Motion Encoder	$\mathbf{Motion}(T, V \times 3) \rightarrow f_m(T, d_m)$	Linear(V × 3, d_m)
Speaker Style Encoder	$\mathbf{MelSpec}(T_{mel}, 80) \rightarrow \tilde{f}_s(d_{txt})$	Conv1d(80, 128) × 8 → [B, T_{mel}, 1280] Conv1d(1280, 128) → [B, T_{mel}, 128] GroupNorm(128/16, 128) Conv1d(128, 128) × 6 → [B, T_{mel}, 128] GroupNorm(128/16, 128) Conv1d(128, 128) × 6 → [B, T_{mel}/8, 128] GroupNorm(128/16, 128) AdaptiveAvgPool1d(1) → [B, 128] Linear(128, 128) × 6 Linear(128, 128) × 6 Linear(128, 256) → [B, 256] Linear(256, d_{txt}) → [B, d_{txt}]
Motion Memory	$f_{txt}(d_{txt}) \rightarrow f_m(d_m)$	Parameter(d_{txt}, d_m)
ASR Encoder	$\mathbf{Audio}(T_a) \rightarrow f_{txt}(T, d_{txt})$	Conv1d(1, d_{ASR}) → [B, T_a, d_{ASR}] LayerNorm(d_{ASR}) LinearInterpolation → [B, T, d_{ASR}] Conv1d(d_{ASR}, d_{ASR}) × 5 → [B, T, d_{ASR}] LayerNorm(d_{ASR}) Transformer(d_{ASR}) Linear(d_{ASR}, d_{txt}) → [B, T, d_{txt}]
Motion Decoder	$f_m(T, d_m + d_{txt}) \rightarrow \mathbf{Motion}(T, V \times 3)$	Linear($d_m + d_{txt}$, d_m) Transformer(d_m) Linear(d_m, $V \times 3$)

Table S5. The detailed architecture of our MemoryTalker.

nunciation.

D. Feature Analysis

D.1. Speaking Style Feature Visualization

We provide visualizations of motion features with t-SNE [42] to show that our method reflects speaking styles in the audio, contrasting it with existing approaches (see Figure S8). Figure S8 (a) visualizes the motion feature synthesized during inference when using one-hot encoding [12]. As in [12], since there are not able to know one-hot class information, one-hot vectors are arbitrarily selected (8 classes). As a result, this model cannot distinguish actual unseen speakers at all. Figure S8 (b) shows the results of utilizing 3D facial mesh sequence [19]. It considers 3D facial mesh sequences as additional inputs. However, the 3D facial meshes are usually unavailable in real-world situations at inference. In addition, it does not distinguish the style distribution corresponding to the unseen speakers. On the other hand, as shown in Figure S8 (c), our model provides clear speakers' clustering that corresponds to individual speaking styles. In particular, the proposed method does not require any prior information (*i.e.*, speaker information) in both training and inference stages, which makes it more practical in real-world scenarios.

D.2. Key Addressing Vector Visualization

To verify which key addressing vector correctly retrieves the motion memory, we visualize the key addressing vectors of our model. Figure S6 shows the generated 3D facial mesh sequences and the corresponding key addressing vectors of MemoryTalker model. The key addressing vectors are generated from an audio segment pertaining to the phoneme "/a:/" and "/w/", respectively, for different speakers. We can see that the address smoothly varies as the lip region in the mesh moves. From visualizing the key addressing vectors of different speakers speaking the same pronunciation, we observe the similar tendency of the key addressing vectors. Focusing on the slots that noticeably change their address value, from the 3rd column, the address on the 8th and 16th slot addresses increases from 0.075 to 0.200 when pronouncing "/a:/", and also when presented with other speakers' speech. Similarly, when pronouncing "/w/", it shows that the address on the 2nd and 17th slot addresses increases regardless of speaker. These demonstrate that the key addressing vectors of our MemoryTalker are activated in the same slot when synthesizing the same lip shape, suggesting that our motion memory slot feature accurately stores the corresponding lip motion. Note that the key addressing vector here is from the ASR model, which is to extract speaker-neutral general mo-

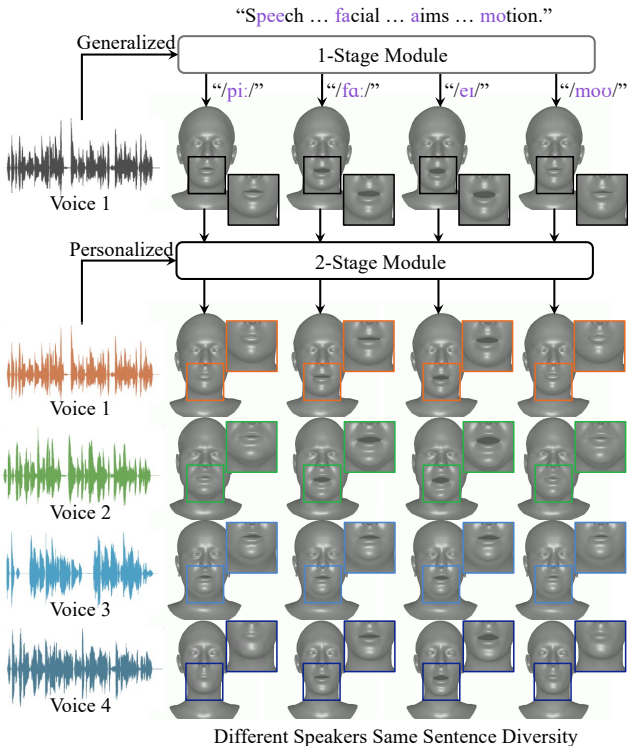

Figure S4. General Motion synchronized with audio is obtained in 1-stage (i.e. w/o 2-stage). Then, Personalized Motion is synthesized in 2-stage using speaking style features extracted from different speakers (i.e. w/ 2-stage).

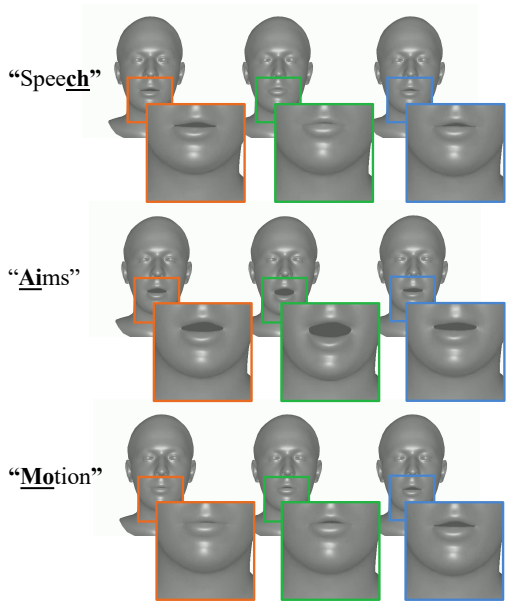

Figure S5. Results of applying different audio inputs to the same identity template. The model generates lip motions synchronized with speech while reflecting speaker-specific styles from audio, all while preserving the target's identity.

tions. Our personalization is further applied to the memory components, not the key addressing vector.

E. Implementation Details

E.1. Details of MemoryTalker Architecture

We configured the main models as in Table S5. This is the baseline architecture used in our all experiments. The Motion Encoder is designed as a single linear layer as in [12, 19, 41, 45]. First, the Speaking Style Encoder concatenates the output and input features of each layer to create a 1280-dimensional feature. After that, it goes through conv1d, group normalization, and ReLU in order, and then learns the features that appear throughout the frame through an average pooling layer. Finally, it is projected as a feature with a dimension of d_{txt} through a linear layer. [8] The Motion Memory uses a text addressing vector with a dimension of d_{txt} as a query and emits a motion feature with a dimension of d_m. The ASR encoder uses the HuBERT [18] structure and learned parameters. First, the audio feature is encoded with a 1D convolution layer and a group normalization layer. The GeLU nonlinear function is used during encoding. After that, linear interpolation is used to match the audio fps with the motion fps (30 fps for VOCASET and 25 fps for BIWI). After that, it passes through a transformer encoder, and then a pre-trained linear layer is used to map the feature to the vocab size. The Motion Decoder reduces the dimension to d_m using a linear layer, combining the retrieved motion feature and the text representation feature, and performs positional encoding. After that, the transformer decoder is used to induce the motion feature that matches the text representation. The obtained motion feature passes through a linear layer and synthesizes a motion that moves a neutral face mesh.

E.2. Detail of User Study

In Figure S9, we attach the user study we provided to the subjects. We evaluated our method against other methods using the user study form used in the faceformer [12] and the mimic [19]. Our qualitative evaluation questionnaire consists of a total of 90 questions. Five questions are to check whether the participant is participating sincerely through the qualification video and to remove outliers. For the remaining 85 questions, we evaluate three qualitative metrics in total to compare the output of our *MemoryTalker* with the outputs of existing SoTA methods and GT for sentences randomly sampled from the 40 sentences.

For the evaluation of Realism, we induce the participant to choose A/B pairs by asking the following question: "Comparing the two full faces, which one looks more realistic?". In this case, participants see the full face in two samples and

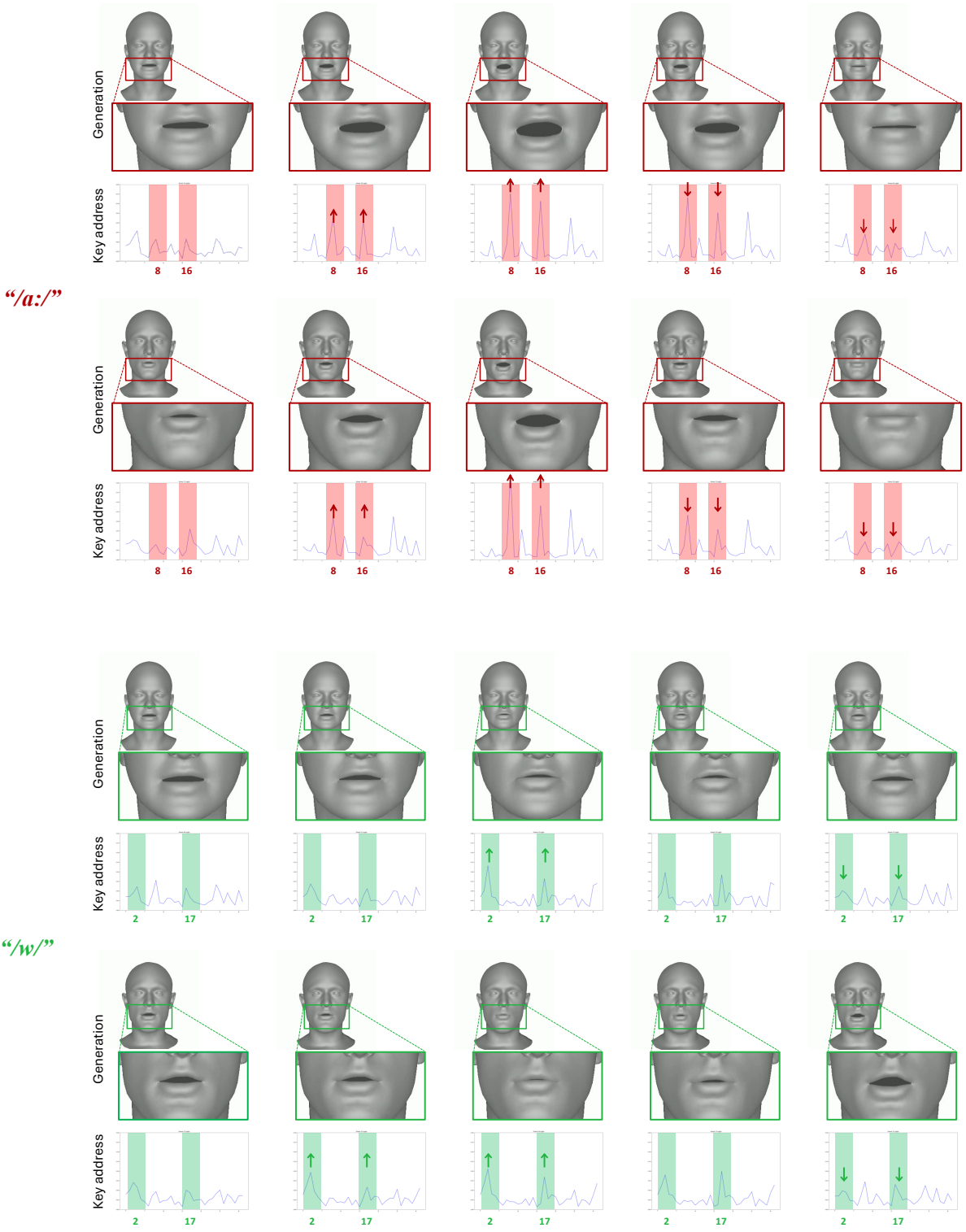

Figure S6. Key addresses from audio input and corresponding generated mesh in a sequence. Note that the key addressing vector here is from the ASR model, which is to extract speaker-neutral general motions. Our personalization is further applied to the memory components afterwards.

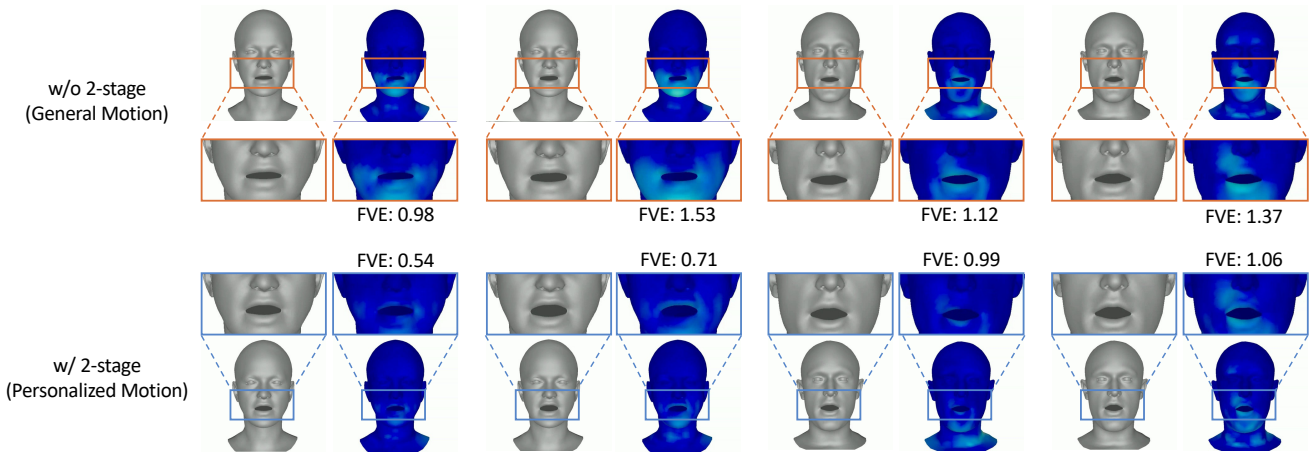

Figure S7. Qualitative comparison of results of learning with 1-stage only (general motion) and those of learning with 2-stage (personalized motion).

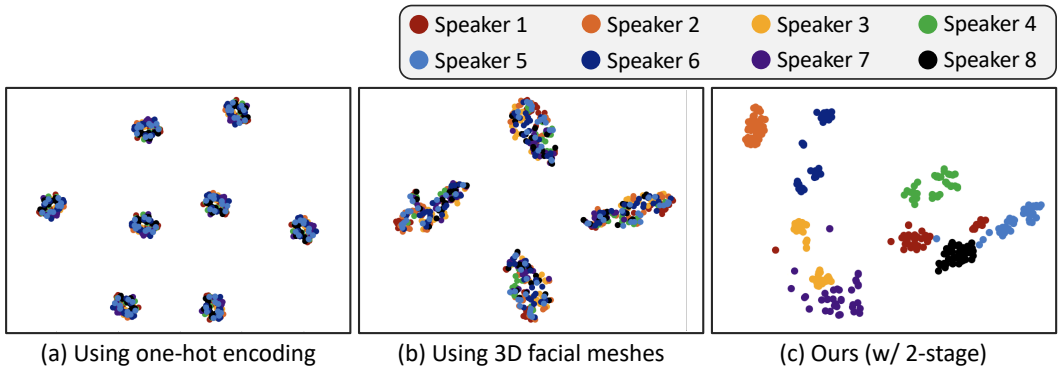

Figure S8. Feature visualization according to different style encoding types: one-hot, 3D facial meshes, and ours.

choose the more natural option.

To measure the Lip-sync, we conducted an A/B test with the question: "Comparing the two lips, which one looks more realistic?". As with evaluating realism, we asked participants to choose the sample that was more synchronized with the audio among the A/B samples. This allowed us to subjectively evaluate the degree of synchronization between the audio and the lip region. To measure how well our *MemoryTalker* captures the speaking style of the ground truth (GT), we compare whether our output is more similar to the GT than the outputs from other models. To this end, we randomly placed the GT video in the first position, and the videos to be compared in the second and third positions. And then, we asked the participant to answer the question "Comparing the speaking style (including the amplitude of mouth opening and closing, the dimensionality of pouting, etc.) of the last two faces, which one is more consistent with the first video?".

Competitors	Lip Sync(%)	Realism(%)
vs. GT	41.1	40.3

Table S6. User study: our method vs. GT on VOCASET [6].

Additionally, to assess participants' reliability in the qualification question, we randomly place the same first-position video in either the second or the third position. At this time, the participant is asked a question about speaking style ("Comparing the speaking style (including the amplitude of mouth opening and closing, the dimensionality of pouting, etc.) of the last two faces, which one is more consistent with the first video?"), and this choice is different from the previous one in that it has a fixed correct answer. Therefore, if the participant answers this question incorrectly, we consider the participant an outlier and remove it from the statistics, thereby improving the quality of our evaluation method.

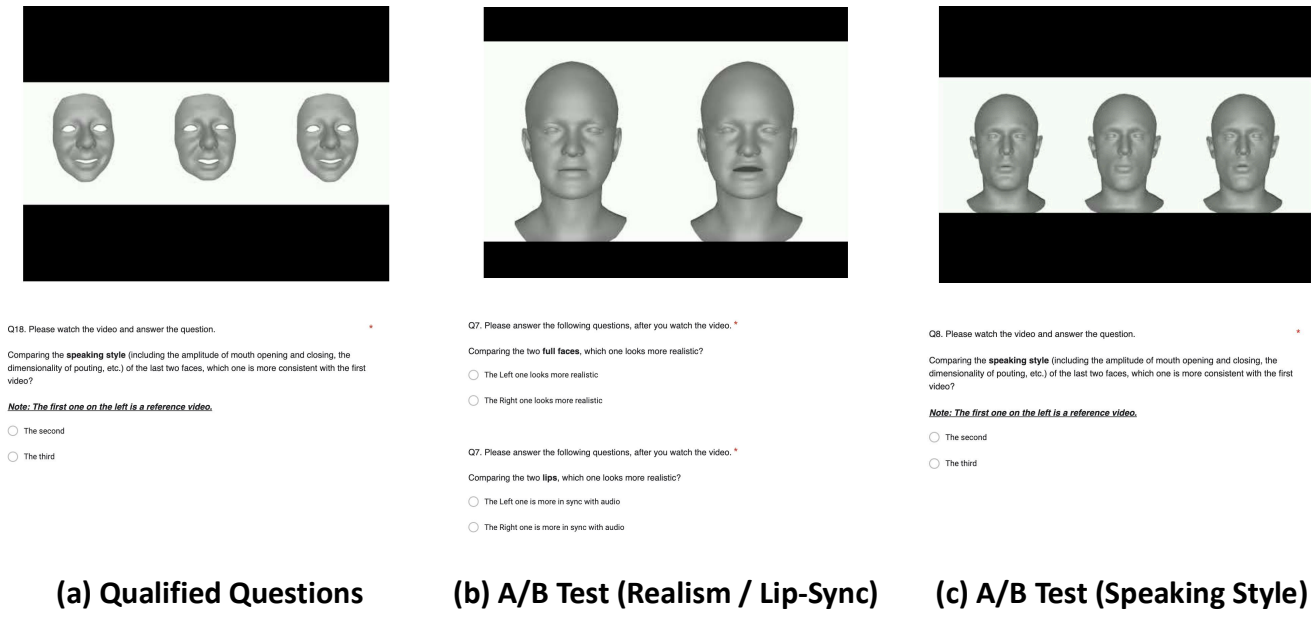

Figure S9. Examples of conducted user study.

In summary, we evaluated three qualitative metrics (Speaking Style, Realism, and Lip-sync) by randomly selecting ten sentences per model, following a similar scale to previous study[12]. Based on the ten sampled videos for each model, we split them into two groups: five videos were used to evaluate Speaking Style, and the remaining five were used to evaluate Realism and Lip-sync. Specifically, we generated one question per video for Speaking Style (5 questions), and two questions per video (one for Realism, one for Lip-sync) for the other five videos (10 questions). Consequently, each model yielded a total of 15 questions (5 + 10) from its ten samples. With five models, we obtained 50 samples in total and generated 90 questions. Additionally, we selected ten more samples to directly compare our method with the ground truth (GT) as S6. Among these, five were used for qualification questions, and the remaining five were used for Realism and Lip-sync evaluations between our approach and the GT. All questions were administered in a forced-choice format, and any participant who answered even one qualification question incorrectly was excluded from the final statistics. Furthermore, we provided two reminders to participants: (1) to ensure their computer's sound was on while watching the videos, and (2) to note that one or two of the videos might be used for qualification purposes, such that random guessing could result in disqualification. Because each video is relatively short (4–7 seconds), we allowed participants to watch them repeatedly up to five times and even replay them additionally, ensuring they could carefully assess each sample. A total of 33 people participated in our user study, and 2 of them were removed because they did not pass the qualification questions we provided. In our experience, the participants took about 40 minutes to complete our user study.