Seeing Speech: Learning Visible Articulatory Dynamics for Speech-Driven 3D Facial Animation

Hyung Kyu Kim **Byungchan Hwang** **Hak Gu Kim**[†]
Chung-Ang University
{hyung1208, byungchan0705, hakgukim}@cau.ac.kr

Abstract

Recent progress in speech-driven 3D facial animation has improved vertex-level reconstruction quality, but speech-consistent visible articulation remains difficult. This is because speech production follows structured and constrained articulators' coordination and the mapping from acoustics to motion is inherently one-to-many. Motivated by the structured patterns of visible articulation, we propose a novel articulation-aware framework that models visible speech through directional articulatory motions and composes them into surface-consistent 3D facial motion. To represent visible articulation with three directional articulatory motions, spreading, opening, and protrusion, we propose a Speech–Articulatory Memory (SAM) that captures the correspondence between speech and these motions under phonetic context through retrieval and decoding based on a key-value memory structure. Then, a Topology-aware Articulatory Composition (TAC) integrates the predicted directional articulatory motions under mesh topology to produce surface-consistent 3D facial motion. Experiments on VOCASET and TFHP show that our method achieves state-of-the-art performance on standard reconstruction metrics and improves visible articulatory distance and velocity errors for lip articulation, while a user study confirms clear preference in lip sync and realism. Project Page: https://irislab.cau.ac.kr/NeurIPS2026-SeeingSpeech/

1 Introduction

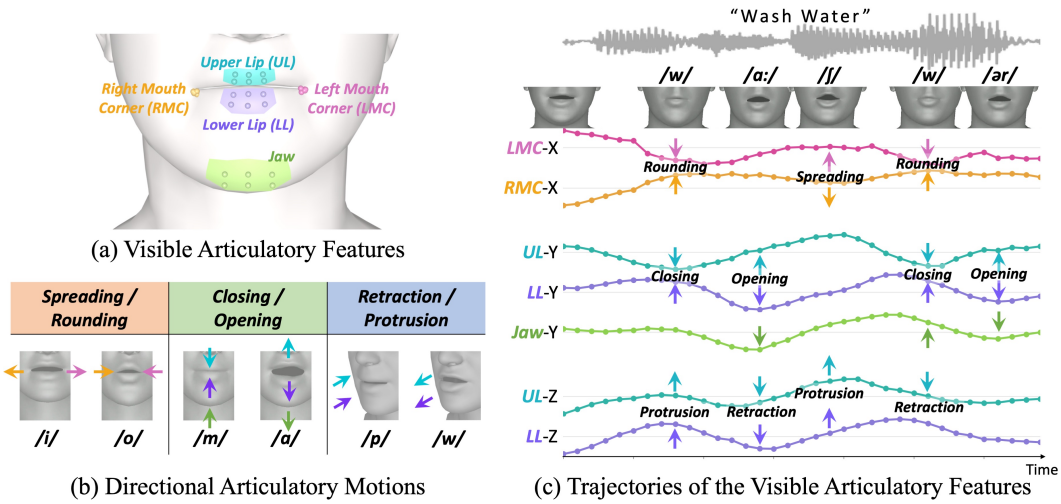
Speech-driven 3D facial animation aims to generate realistic 3D facial motion from speech. This enables applications such as VR, education, and speech training with visualized articulation [1–3]. However, generating speech-consistent 3D facial motion is challenging due to one-to-many mapping between acoustics and facial motion [4–6]. This stems from articulatory coupling and compensatory mechanisms, where multiple articulator configurations under structured articulatory constraints can produce the same acoustic realization [7, 8]. Therefore, modeling these articulatory characteristics is important for generating speech-consistent facial motion.

Despite this, most existing speech-driven facial animation methods do not explicitly model articulatory motion constraints. Recent approaches, including direct mapping with temporal CNNs [9, 10], Transformer-based auto-regressive prediction [11, 12], and diffusion-based denoising frameworks [13–17], primarily focus on improving reconstruction quality at the vertex or mesh level. By formulating speech-driven animation as a holistic face-motion regression problem, these methods synthesize overall 3D facial trajectories without explicitly accounting for the structured articulatory motions underlying speech production [18, 19]. As a result, holistic vertex regression offers limited interpretability and weak grounding in speech articulation.

This raises an intriguing research question: *Can we model speech-driven 3D facial animation in a way that is grounded in the visible articulation of human speech production?* Human speech is generated

[†]Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



(a) Visible Articulatory Features

(b) Directional Articulatory Motions

(c) Trajectories of the Visible Articulatory Features

Figure 1: How do we model articulation-grounded facial motion from speech? (a) Five visible mesh-surface anchors as motion reference points: upper lip (UL), lower lip (LL), left mouth corner (LMC), right mouth corner (RMC), and Jaw. (b) From these anchors, we describe visible articulation with three directional articulatory motions: *horizontal* rounding/spreading, *vertical* opening/closing, and *depth-wise* protrusion/retraction. (c) For an utterance, the trajectories of these motions form temporally localized articulatory events that align with acoustic/phonemic cues.

through coordinated, time-varying configurations of articulators [20]. The lips and jaw are externally observable and play a central role in shaping intelligible speech. This suggests that speech-consistent facial animation should be grounded in structured visible articulatory motions [7, 21, 22]. Motivated by this perspective, we examine how *visible articulation* evolves during speech production. *Visible articulation* refers to the observable movements and configurations of speech organs, particularly the lips and jaw, in producing speech [18]. As illustrated in Fig. 1 (a) and (b), we define three directional articulatory motions (i.e., Spreading–Rounding, Opening–Closing, and Protrusion–Retraction) based on mesh-surface anchors (UL, LL, LMC, RMC, and Jaw), which specify the directions of lip–jaw articulations [23, 24]. Across representative phonemes, we observe that visible articulation is not arbitrary, but follows consistent, directionally dominant motion patterns (see Fig. 1 (c)).

In this paper, we propose a novel speech-driven 3D facial animation framework grounded in the structured nature of visible articulation during human speech production. Based on this observation, we interpret speech-driven facial motion as arising from directionally organized articulatory motions that recur across phonemes and reflect consistent visible articulatory coordination patterns. For this purpose, we first introduce a Speech–Articulatory Memory (SAM) to capture the correspondence between speech and directional articulatory motions, modeling how speech gives rise to specific patterns of visible articulation under phonetic context. Second, we propose a Topology-aware Articulatory Composition (TAC) to translate directional articulatory motions into surface-consistent 3D facial motion, ensuring that visible articulation is realized consistently over the facial surface under the mesh topology. By grounding speech-driven facial animation in structured visible articulation and its directional organization, our approach emphasizes interpretability and consistency with the articulatory structure of human speech production.

Our main contributions are summarized as follows:

- To the best of our knowledge, we propose the first *articulation*-aware framework for speech-driven 3D facial animation grounded in structured visible articulation.
- We propose a novel memory network named Speech–Articulatory Memory (SAM) to model the relationship between speech and directional articulatory motions. SAM captures phonetic context-dependent visible articulation through memory-based retrieval and decoding.
- We propose a Topology-aware Articulatory Composition (TAC) to compose directional articulatory motions into surface-consistent 3D facial motion. TAC ensures consistent visible articulation over the facial surface under mesh topology.
- With quantitative and qualitative evaluations, including user studies, we demonstrate the effectiveness of SAM and TAC. Our method shows superior performance and improved interpretability over state-of-the-art speech-driven 3D facial animation models.

2 Related Works

2.1 Speech-Driven 3D Facial Animation

Speech-driven 3D facial animation synthesizes 3D facial motion from speech [10, 25–35]. Early methods adopt temporal CNNs [9] or Transformers [11] to model temporal dependencies. Subsequent methods improve motion fidelity with discrete motion priors [12, 36, 37], key-motion priors [38], and lip-reading supervision [39, 40]. Beyond motion fidelity, UniTalker [41] trains a single model on datasets with heterogeneous topologies [9, 42–44], and ScanTalk [45] removes dependence on a fixed mesh topology. For richer animation, recent methods incorporate pose [14, 46], style [47–50], or emotion [43, 51–61].

Although these approaches mainly improve motion fidelity, topology handling, or controllability within audio-to-face generation, they do not explicitly model the visible articulatory structure underlying speech production. Prior articulatory studies show that visible speech exhibits lip–jaw motion patterns, including lip opening, horizontal spreading or rounding, and protrusion or retraction [23, 24, 62, 63]. Motivated by this observation, we represent visible articulation with three directional articulatory motions and compose them over the mesh topology to produce 3D facial motion, grounding generation in the visible articulation of speech production [18, 19].

2.2 Articulatory Modeling for Speech and Animation

Visible articulators such as the lips and jaw exhibit phoneme-dependent and spatially localized motion patterns, in which specific phonemes are associated with dominant motions in lip opening or closing, horizontal spreading or rounding, or depth-wise protrusion or retraction [23, 24, 64, 65]. Such lip and jaw movements, along with other articulatory motions, can be predicted from speech [66–68]. Building on these insights, several animation methods incorporate articulatory structure through parameterized representations. JALI [69] decomposes facial motion into lip and jaw action parameters [70]. VisemeNet [71] predicts these parameters from audio. In such approaches, each phoneme is associated with a predefined viseme, and facial motion is generated by blending these fixed shapes. While this provides a structured control space, articulation is reduced to a low-dimensional parameter set, where context-dependent variation is expressed through combinations of predefined viseme patterns. More broadly, FACS [72] is a widely adopted coding system for describing observable facial actions and is often used as the basis for facial rigs. FACS-style controls are named, unidirectional units, so each articulatory degree of freedom is split across several of them. For example, opening and closing correspond to separate jaw-opening and mouth-opening controls, and spreading and rounding are split between mouth stretch and mouth pucker. FLAME [73], in contrast, is highly expressive, yet its expression parameters are unnamed PCA coefficients, making individual coefficients difficult to interpret in terms of speech articulation. Blendshapes emphasize intuitive control, and FLAME emphasizes expressive parameterization. These representations were not designed to provide explicit descriptions of articulatory motion.

In natural speech, articulatory motion is not expressed as combinations of fixed shapes or unnamed PCA coefficients, but as continuous changes in vertex positions that vary with phonetic context. To address these limitations, we move beyond viseme-based and FACS-style parameterization and model speech-driven facial motion directly at the vertex level. Specifically, we represent articulation through three canonical directions characterized in the speech production literature [64] and predict how speech drives these direction-specific motions across the facial surface.

3 Proposed Method

Fig. 2 shows an overview of our articulation-aware framework. Given a facial template mesh with V vertices, we represent visible articulation through three directional articulatory motions grounded in speech production: horizontal spreading (S), vertical opening (O), and depth-wise protrusion (P) [64, 74]. We model facial motion $\mathbf{v}^t \in \mathbb{R}^{V\times 3}$ at time t through these three directions, corresponding to the x-, y-, and z-components. To generate articulation-grounded 3D facial animation, the proposed method consists of two main parts, which are Speech–Articulatory Memory (SAM) and Topology-aware Articulatory Composition (TAC). First, SAM retrieves directional articulatory motion features $\mathbf{f}_{\mathrm{retri},\delta}^{t-\Omega:t+\Omega}$ from value memories $\mathbf{M}_{\mathrm{val},\delta}$ using an acoustic query $\mathbf{q}^t$ and an acoustic

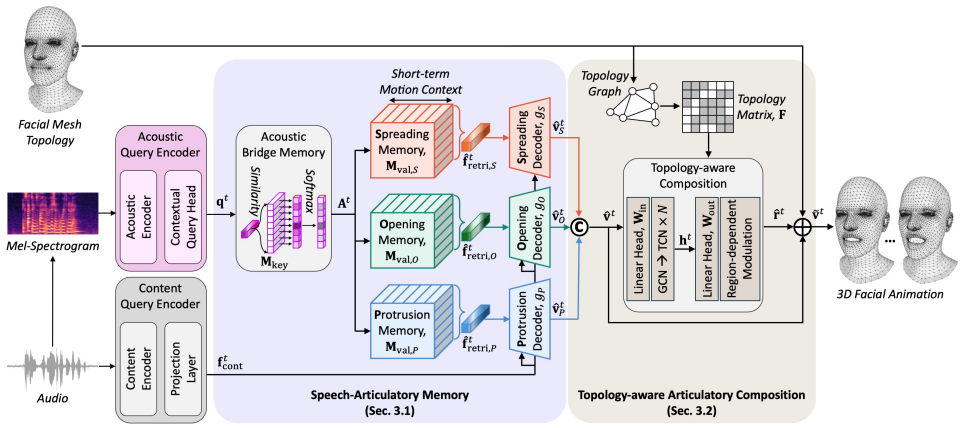


Figure 2: Overall architecture of our articulation-aware framework. SAM retrieves directional articulatory motion features (Spreading–Rounding, Opening–Closing, Protrusion–Retraction) from speech. TAC composes them into surface-consistent 3D facial motion under the mesh topology.

bridge memory $\mathbf{M}_{\text{key}}$, and refined via cross-attention with content features $\mathbf{f}^t_{\text{cont}}$ to predict directional articulatory motion $\hat{\mathbf{v}}^t_\delta$ for each direction δ. Second, TAC composes the refined motions under the mesh topology to produce surface-consistent 3D facial motion $\tilde{\mathbf{v}}^t$.

3.1 Speech–Articulatory Memory (SAM)

Acoustic-to-Articulatory Motion Memory Addressing. Acoustic bridge memory $\mathbf{M}_{\text{key}} \in \mathbb{R}^{N\times D_{\text{SAM}}}$ (N learnable slots of dimension D_{SAM}) is designed to associate speech acoustics with articulatory motion patterns. We first encode the mel-spectrogram into an acoustic query $\mathbf{q}^t \in \mathbb{R}^{1\times D_{\text{SAM}}}$ using an acoustic encoder with a contextual query head that aggregates a temporal window of size $2\Omega+1$ around time t to capture context-dependent phonetic cues [19]. We then compute the addressing vector $\mathbf{A}^t \in \mathbb{R}^{1\times N}$ via cosine similarity followed by scaled softmax with τ_{SAM}=16.0.

$$\mathbf{A}^t_j = \frac{\exp(s^t_j \cdot \tau_{\text{SAM}})}{\sum_{k=1}^{N}\exp(s^t_k \cdot \tau_{\text{SAM}})}, \quad \text{where } s^t_j = \frac{\mathbf{q}^t \mathbf{M}^{\top}_{\text{key},j}}{\|\mathbf{q}^t\|_2 \cdot \|\mathbf{M}_{\text{key},j}\|_2}, \quad j = 1,\dots,N. \tag{1}$$

Directional Articulatory Motion Retrieval. Given the addressing vector $\mathbf{A}^t$, we retrieve directional motion features from three learnable value memories $\mathbf{M}_{\text{val},\delta} \in \mathbb{R}^{N\times(2\Omega+1)\times D_{\text{SAM}}}$ corresponding to *Spreading* (S), *Opening* (O), and *Protrusion* (P). Each memory slot stores a directional motion feature sequence of length $2\Omega+1$, encoding short-term temporal continuity. For each articulatory direction δ, a motion feature sequence $\mathbf{f}^{t-\Omega:t+\Omega}_{\text{retri},\delta} \in \mathbb{R}^{(2\Omega+1)\times D_{\text{SAM}}}$ is retrieved from the corresponding value memory using the addressing vector $\mathbf{A}^t$ via weighted summation, which can be written as

$$\mathbf{f}^{t-\Omega:t+\Omega}_{\text{retri},\delta} = \sum_{j=1}^{N} \mathbf{A}^t_j \cdot \mathbf{M}_{\text{val},\delta,j}, \tag{2}$$

where Ω is the half-window defining the temporal span of local phonetic context. The retrieved sequence $\mathbf{f}^{t-\Omega:t+\Omega}_{\text{retri},\delta}$ represents short-term articulatory motion features.

Content-guided Motion Refinement. To disambiguate phoneme-dependent articulatory patterns, we refine the retrieved motion features by conditioning on content representations from a self-supervised speech model. Following [32], we extract a content feature $\mathbf{f}^t_{\text{cont}} \in \mathbb{R}^{1\times D_{\text{SAM}}}$ for $t = 1,\cdots,T$ using a content encoder [75]. The extracted features are resampled to match the dataset frame rate and then projected via a linear layer. Using the content feature sequence $\mathbf{f}^{1:T}_{\text{cont}}$ as the query and $\hat{\mathbf{f}}^{1:T}_{\text{retri},\delta}$, summarized from retrieved sequences, as key/value, a per-direction Transformer decoder $g_\delta(\cdot)$ performs temporal cross-attention to produce the directional articulatory motion sequence $\hat{\mathbf{v}}^{1:T}_\delta$.

$$\hat{\mathbf{v}}^{1:T}_\delta = g_\delta(\mathbf{f}^{1:T}_{\text{cont}}, \hat{\mathbf{f}}^{1:T}_{\text{retri},\delta}) = \text{softmax}\left(\frac{\mathbf{Q}_\delta \mathbf{K}^{\top}_\delta}{\sqrt{D_{\text{SAM}}}}\right)\mathbf{V}_\delta\,\mathbf{W}_{\text{out},\delta}, \tag{3}$$

where $\mathbf{Q}_\delta = (\mathbf{f}^{1:T}_{\text{cont}} + \mathbf{P})\mathbf{W}_{Q,\delta}$ with periodic positional encoding $\mathbf{P}$ of period ρ to encode temporal periodicity (see Appendix B.2 for details), $\mathbf{K}_\delta = \hat{\mathbf{f}}^{1:T}_{\text{retri},\delta}\mathbf{W}_{K,\delta}$, and $\mathbf{V}_\delta = \hat{\mathbf{f}}^{1:T}_{\text{retri},\delta}\mathbf{W}_{V,\delta}$. Here,

$\mathbf{W}_{Q,\delta}, \mathbf{W}_{K,\delta}, \mathbf{W}_{V,\delta} \in \mathbb{R}^{D_{\text{SAM}} \times D_{\text{SAM}}}$ and $\mathbf{W}_{\text{out},\delta} \in \mathbb{R}^{D_{\text{SAM}} \times V}$ are learnable projection matrices. $\hat{\mathbf{f}}^t_{\text{retri},\delta} = \sum_{\omega=-\Omega}^{+\Omega} w_\delta(\omega) \cdot \mathbf{f}^{t+\omega}_{\text{retri},\delta}$ summarizes the retrieved sequence via learnable weights $w_\delta(\omega)$.

Training Objectives. We train the Speech–Articulatory Memory (SAM) to retrieve and synthesize directional articulatory motion $\hat{\mathbf{v}}^t_\delta$ by aligning with the corresponding ground-truth motion $\mathbf{v}^t_\delta$. To this end, we apply a directional reconstruction loss $\mathcal{L}_{\text{rec},\delta}$ to enforce accurate prediction of the ground-truth motion $\mathbf{v}^t_\delta$ in the vertex domain. In addition, a directional velocity loss $\mathcal{L}_{\text{vel},\delta}$ on temporal differences $\Delta\mathbf{v}^t_\delta$ is also employed to encourage the predicted motion to follow the correct temporal evolution. These complementary supervisions jointly enforce accurate spatial reconstruction and temporally consistent motion dynamics for each articulatory direction δ.

$$\mathcal{L}_{\text{rec},\delta} = \frac{1}{T}\sum_{t=1}^{T} \|\mathbf{v}^t_\delta - \hat{\mathbf{v}}^t_\delta\|_2^2, \quad \mathcal{L}_{\text{vel},\delta} = \frac{1}{T-1}\sum_{t=1}^{T-1} \|\Delta\mathbf{v}^t_\delta - \Delta\hat{\mathbf{v}}^t_\delta\|_2^2, \tag{4}$$

where $\Delta\mathbf{v}^t_\delta = \mathbf{v}^{t+1}_\delta - \mathbf{v}^t_\delta$. These supervisions mitigate shortcut compensation across directions and encourage each directional motion to follow its intended spatial and temporal behavior.

We further introduce a value memory store loss $\mathcal{L}_{\text{store},\delta}$ to guide the value memory to store temporally coherent articulatory motion patterns. For this purpose, we align the retrieved motion feature sequence $\mathbf{f}^{t-\Omega:t+\Omega}_{\text{retri},\delta}$ with the reference motion feature sequence $\mathbf{z}^{t-\Omega:t+\Omega}_\delta$ defined over a $(2\Omega+1)$-frame temporal window. For each articulatory direction δ, the reference motion feature $\mathbf{z}^t_\delta \in \mathbb{R}^{1\times D_{\text{SAM}}}$ is extracted from ground-truth motion $\mathbf{v}^t_\delta$ using a pre-trained per-direction autoencoder, which is trained to encode $\mathbf{v}^t_\delta$ into $\mathbf{z}^t_\delta$ and reconstruct $\mathbf{v}^t_\delta$ from $\mathbf{z}^t_\delta$ (see Appendix B.1 for details). At each time step t, we align the reference window $\mathbf{z}^{t-\Omega:t+\Omega}_\delta$ with the $\mathbf{f}^{t-\Omega:t+\Omega}_{\text{retri},\delta}$ via cosine distance:

$$\mathcal{L}_{\text{store},\delta} = \frac{1}{T\cdot(2\Omega+1)}\sum_{t=1}^{T}\sum_{\omega=-\Omega}^{+\Omega}\left(1 - \frac{\mathbf{f}^{t+\omega}_{\text{retri},\delta}\cdot\mathbf{z}^{t+\omega}_\delta}{\|\mathbf{f}^{t+\omega}_{\text{retri},\delta}\|_2\|\mathbf{z}^{t+\omega}_\delta\|_2}\right). \tag{5}$$

The window-level alignment encourages the value memory to capture short-term temporal consistency in articulatory motion, reducing one-to-many ambiguity in acoustic-to-motion mapping.

The total loss for SAM is $\mathcal{L}_{\text{SAM}} = \sum_{\delta\in\{S,O,P\}}(\mathcal{L}_{\text{rec},\delta} + \mathcal{L}_{\text{vel},\delta} + \lambda_1\mathcal{L}_{\text{store},\delta})$, with $\lambda_1 = 10^{-6}$.

3.2 Topology-aware Articulatory Composition (TAC)

In TAC, the predicted articulatory motions $\hat{\mathbf{v}}^t_\delta$ are composed into surface-consistent 3D facial motion by considering their spatio-temporal coordination over the facial surface. To this end, we introduce a topology-aware composition that captures spatial coupling across vertices and temporal consistency.

Spatio-Temporal Composition via Mesh Topology. For spatial coordination, we leverage the facial topology adjacency matrix $\mathbf{F} \in \mathbb{R}^{V\times V}$ derived from the facial template mesh. $\mathbf{F}_{ij} = 1$ if vertices i and j are connected by an edge, and $\mathbf{F}_{ij} = 0$ otherwise. Let $\mathbf{h}^t \in \mathbb{R}^{V\times D_{\text{TAC}}}$ denote the topology-aware motion feature. We first stack the refined articulatory motions $\hat{\mathbf{v}}^t_\delta$ for each direction δ as $\hat{\mathbf{v}}^t = [\hat{\mathbf{v}}^t_S,\ \hat{\mathbf{v}}^t_O,\ \hat{\mathbf{v}}^t_P] \in \mathbb{R}^{V\times 3}$ and project it into a latent space $\hat{\mathbf{h}}^t = \hat{\mathbf{v}}^t\mathbf{W}_{\text{in}} \in \mathbb{R}^{V\times D_{\text{TAC}}}$, where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{3\times D_{\text{TAC}}}$. Then, $\mathbf{h}^t$ can be obtained by propagating motion information across neighboring vertices with a graph convolutional layer [76].

$$\mathbf{h}^t \;=\; \sigma\left(\hat{\mathbf{F}}\cdot\hat{\mathbf{h}}^t\cdot\mathbf{W}\right), \tag{6}$$

where $\hat{\mathbf{F}} = \tilde{\mathbf{D}}^{-1/2}(\mathbf{F}+\mathbf{I})\tilde{\mathbf{D}}^{-1/2}$ is the symmetrically normalized adjacency matrix following [76] and $\sigma(\cdot)$ is a non-linear activation. $\tilde{\mathbf{D}}$ is the degree matrix of $\mathbf{F}+\mathbf{I}$, and $\mathbf{W} \in \mathbb{R}^{D_{\text{TAC}}\times D_{\text{TAC}}}$ is a learnable transformation. This operation injects mesh-topology information into $\mathbf{h}^t$, allowing each vertex representation to reflect coordinated motions over its topological neighborhood.

To capture temporal dynamics, we model the topology-aware motion features as vertex-wise time-varying sequences and apply a temporal convolutional network (TCN) along the time axis. For each vertex i, the topology-aware motion feature sequence $\mathbf{h}^{1:T}_i \in \mathbb{R}^{T\times D_{\text{TAC}}}$ is defined as

$$\mathbf{h}^{1:T}_i \;\leftarrow\; \text{TCN}\left(\mathbf{h}^{1:T}_i\right), \quad i = 1,\ldots,V. \tag{7}$$

This temporal composition allows each vertex to capture short-term temporal dependencies while preserving spatial coordination from the mesh topology, resulting in consistent facial motion. Finally, we project the topology-aware motion feature sequences $\mathbf{h}^{1:T}$ to the vertex domain via a linear layer $\mathbf{W}_{\text{out}} \in \mathbb{R}^{D_{\text{TAC}} \times 3}$ yielding topology-aware motion residual $\mathbf{r}^t = \mathbf{h}^t \mathbf{W}_{\text{out}} \in \mathbb{R}^{V \times 3}$.

Region-adaptive Motion Modulation. Facial regions exhibit different deformation characteristics (e.g., highly deformable lips/jaw vs. rigid cheeks/forehead). To reflect such region-dependent deformability, we introduce a learnable per-vertex modulation gate $\mathbf{m} \in \mathbb{R}^{V \times 1}$, where $\mathbf{m} = V \cdot \text{softmax}(\tau_{\text{TAC}} \cdot \boldsymbol{\alpha})$. Here, $\boldsymbol{\alpha} \in \mathbb{R}^{V \times 1}$ is a learnable and τ_{TAC}=16.0 is a temperature constant. We broadcast $\mathbf{m}$ along the coordinate dimension to obtain a region-modulated motion residual.

$$\hat{\mathbf{r}}^t = (\mathbf{m} \cdot \mathbb{1}) \odot \mathbf{r}^t, \tag{8}$$

where $\mathbb{1} \in \mathbb{R}^{1 \times 3}$ is an all-ones vector and $\odot$ denotes element-wise multiplication and $\mathbf{m}$ modulates $\mathbf{r}^t$ region-adaptively, allowing different facial regions to absorb the topology-aware motion residual with different sensitivities. Finally, the surface-consistent 3D facial motion is obtained as $\tilde{\mathbf{v}}^t = \hat{\mathbf{v}}^t + \hat{\mathbf{r}}^t$.

Training Objectives. We train the TAC to learn a topology-aware refinement that composes the directional articulatory motions into surface-consistent 3D facial motion. $\mathbf{v}^t \in \mathbb{R}^{V \times 3}$ denotes the ground-truth facial motion field at t. Let $\mathbf{L} = \mathbf{I} - \hat{\mathbf{F}}$ denote the normalized graph Laplacian derived from the mesh topology. The reconstruction, velocity, and graph Laplacian losses can be defined as

$$\mathcal{L}_{\text{rec}} = \frac{1}{T} \sum_{t=1}^{T} \|\mathbf{v}^t - \tilde{\mathbf{v}}^t\|_2^2, \quad \mathcal{L}_{\text{vel}} = \frac{1}{T-1} \sum_{t=1}^{T-1} \|\Delta \mathbf{v}^t - \Delta \tilde{\mathbf{v}}^t\|_2^2, \quad \mathcal{L}_{\text{lap}} = \frac{1}{TV} \sum_{t=1}^{T} \|\mathbf{L}\mathbf{v}^t - \mathbf{L}\tilde{\mathbf{v}}^t\|_2^2, \tag{9}$$

While $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{vel}}$ enforce per-vertex spatial and temporal accuracy, $\mathcal{L}_{\text{lap}}$ measures the deviation of each vertex from the topology-weighted average of its neighbors, enforcing local surface consistency. The total TAC loss is $\mathcal{L}_{\text{TAC}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{vel}} + \lambda_2 \mathcal{L}_{\text{lap}}$, with $\lambda_2 = 10^3$.

4 Experiments

4.1 Datasets

We evaluate on two widely used datasets, VOCASET [9] and TFHP [14]. Dataset details and preprocessing are provided in Appendix B.3.

4.2 Implementation Details

All experiments are conducted on a single NVIDIA RTX 4090 GPU. We adopt a two-stage training strategy with the Adam optimizer (learning rate 10^{-4}, batch size 1). In Stage 1, we train SAM for 250 epochs with the Wav2Lip [77] mel-spectrogram encoder and pre-trained HuBERT [75] as the content encoder. In Stage 2, we freeze SAM and train TAC for 150 epochs. We use N=32 memory slots, Ω=7, D_{SAM}=128, and D_{TAC}=64. $\hat{\mathbf{F}}$ is pre-computed from the dataset-specific mesh topology and kept fixed during training. We apply zero padding for temporal boundary handling. Architecture details are provided in Appendix B.1.

4.3 Evaluation Metrics

We use four standard metrics: **FVE** (mean Euclidean distance over all vertices), **LVE** (per-frame max ℓ_2 error on lip vertices) [11], **FDD** (upper-face dynamics deviation) [12], and **LDTW** (DTW [78]-based lip alignment) [47].

In addition, we introduce visible articulatory distance metrics computed from lip regions shown in Fig. 1(a). Let $\mathbf{p}_{\text{UL}}(t)$, $\mathbf{p}_{\text{LL}}(t)$, $\mathbf{p}_{\text{LMC}}(t)$, and $\mathbf{p}_{\text{RMC}}(t) \in \mathbb{R}^3$ denote the region-mean 3D positions of UL, LL, LMC, and RMC at frame t, respectively. We define three articulatory measures as

$$\begin{aligned} \text{ICW}(t) &= |\mathbf{p}_{\text{LMC},x}(t) - \mathbf{p}_{\text{RMC},x}(t)|, \\ \text{ILD}(t) &= |\mathbf{p}_{\text{UL},y}(t) - \mathbf{p}_{\text{LL},y}(t)|, \\ \text{LP}(t) &= (\mathbf{p}_{\text{UL},z}(t) - \mathbf{p}^{\text{neu}}_{\text{UL},z}) + (\mathbf{p}_{\text{LL},z}(t) - \mathbf{p}^{\text{neu}}_{\text{LL},z}), \end{aligned} \tag{10}$$

where ICW measures the inter-commissural width (lip spreading) and ILD measures the inter-labial distance (lip opening). LP captures the lip protrusion relative to the neutral template positions $\mathbf{p}^{\text{neu}}_{\text{UL},z}$ and $\mathbf{p}^{\text{neu}}_{\text{LL},z}$. We evaluate articulation accuracy using **Visible Articulatory Distance Error** (RMSE on distance) and **Visible Articulatory Velocity Error** (RMSE on temporal derivatives).

4.4 Quantitative Results

4.4.1 Overall Comparisons

As shown in Table 1, our method achieves the lowest FVE, LVE, and LDTW on both datasets. The LVE and LDTW gains indicate that articulatory modeling improves lip-region reconstruction, while the lowest FVE demonstrates that overall reconstruction quality is preserved. Results on the TFHP demonstrate consistent lip-region improvements without degrading overall reconstruction quality across various speaking styles. But our method does not explicitly consider upper-face dynamics for its impact on FDD (see Appendix H).

Table 1: Quantitative results on VOCASET and TFHP.

Dataset	Methods	FVE↓	LVE↓	FDD↓	LDTW↓
VOCASET	FaceFormer	0.878	0.346	0.143	0.186
	CodeTalker	0.913	0.397	0.160	0.204
	SelfTalk	0.844	0.305	0.148	0.174
	Imitator	0.898	0.384	0.179	0.199
	ScanTalk	0.878	0.310	0.152	0.168
	UniTalker	0.820	0.282	<u>0.101</u>	0.188
	MemoryTalker	0.810	<u>0.250</u>	0.127	<u>0.151</u>
	StreamingTalker	<u>0.782</u>	0.272	0.191	0.169
	Ours	**0.771**	**0.235**	**0.097**	**0.132**
TFHP	Mimic	<u>0.106</u>	<u>0.116</u>	0.619	<u>0.329</u>
	DiffPoseTalk	0.176	0.159	**0.321**	0.382
	ARTalk	0.206	0.164	0.581	0.385
	Ours	**0.100**	**0.108**	<u>0.578</u>	**0.318**

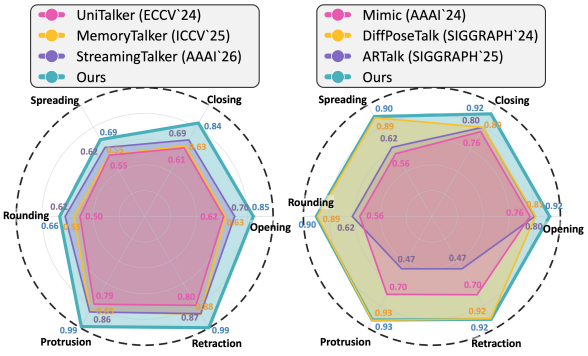
4.4.2 Articulatory Motion Comparisons

To measure articulatory motion, we use RMSE-based distance and velocity errors on three articulatory measures, which are ICW, ILD, and LP. As shown in Table 2, our method achieves the lowest distance errors on all three measures, indicating that the directional decomposition accurately captures the spatial magnitude of articulatory motion. For velocity errors, our method performs the best on ILD and LP and the second-best on ICW, suggesting that this spatial accuracy is accompanied by consistent temporal dynamics. Additional results are provided in Appendix E.2.

Table 2: Visible articulatory distance and velocity errors on VOCASET.

Methods	Distance Error			Velocity Error		
	ICW↓	ILD↓	LP↓	ICW↓	ILD↓	LP↓
FaceFormer	1.764	4.558	2.429	0.581	1.850	1.230
CodeTalker	1.725	4.777	2.808	0.552	2.009	1.343
SelfTalk	1.680	4.031	2.470	0.544	1.958	1.275
Imitator	1.731	4.851	2.526	0.571	1.873	1.205
ScanTalk	1.737	4.034	2.485	0.558	1.854	1.285
UniTalker	<u>1.585</u>	3.887	2.532	0.541	<u>1.627</u>	1.211
MemoryTalker	1.626	<u>3.679</u>	<u>2.347</u>	0.560	1.813	1.224
StreamingTalker	1.658	3.955	2.478	**0.507**	1.645	<u>1.187</u>
Ours	**1.568**	**3.407**	**2.346**	<u>0.521</u>	**1.613**	**1.177**

Beyond per-frame accuracy, we evaluate whether each method preserves the relative magnitude of directional articulatory motions. For each direction δ, we derive a 1D kinematic trajectory from the lip anchors (UL, LL, LMC, RMC) and compute its temporal slope. We decompose the slope into positive and negative components, corresponding to paired motions (e.g., Opening–Closing), and measure the magnitude of each motion as the mean absolute value of its component over time. We then compare the predicted and ground-truth magnitudes using $\exp(-|\log(r)|)$, where r is their ratio (equal to 1 when matched, decaying symmetrically otherwise). See Appendix C.3 for details.



(a) VOCASET (b) TFHP

Figure 3: Radar plot of directional lip-motion tendency strength relative to ground truth.

Fig. 3 shows the results as radar plots normalized by ground truth. Our method remains closest to or on par with the ground truth across all directions, indicating more balanced directional articulation.

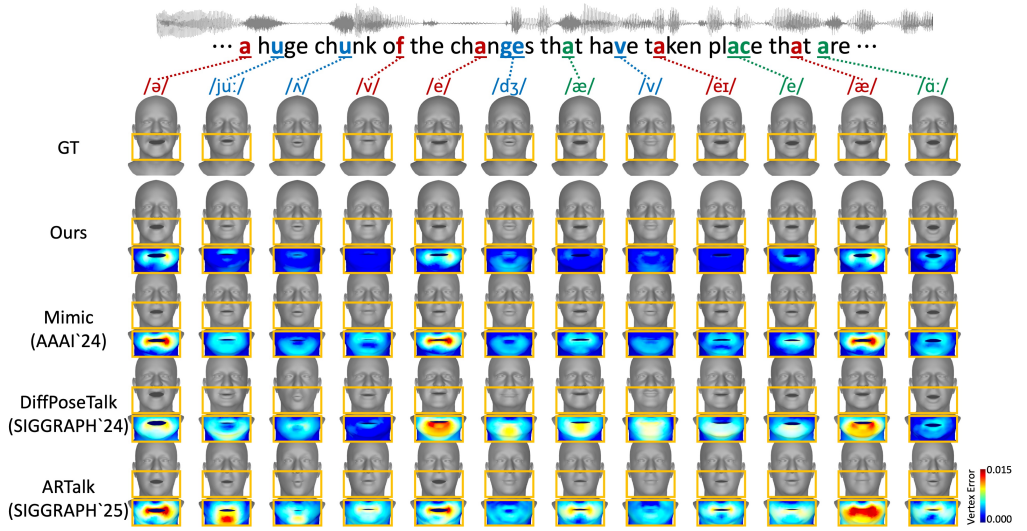
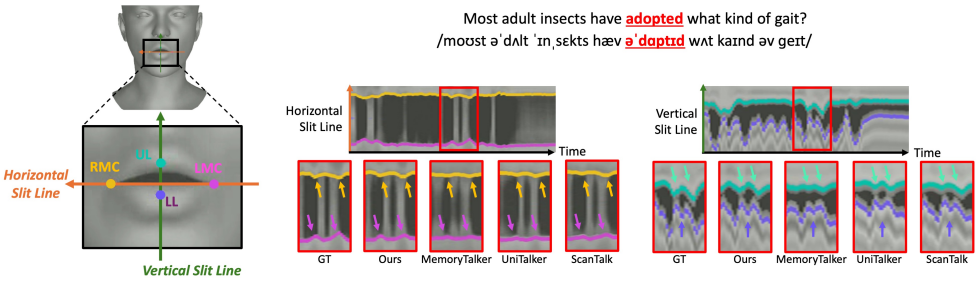


Figure 4: Sequence-level qualitative results on TFHP. For each method, mesh renderings (top) and per-vertex L1 error against the ground truth (bottom, color-coded from 0 (blue) to 0.015 (red)) are shown over time at phoneme-aligned frames.



(a) Visualization of Slit Lines (b) Horizontal Slit-scan Image (c) Vertical Slit-scan Image

Figure 5: Slit-scan visualizations for sequence-level comparison. (a) lip ROI and landmarks (UL, LL, LMC, RMC) defining horizontal (LMC–RMC) and vertical (UL–LL) slit lines. (b) Horizontal and (c) Vertical slit-scans over time, reflecting lip spreading (width) and lip opening (height), respectively.

4.5 Qualitative Results

4.5.1 Sequence-Level Mesh Comparisons

Fig. 4 shows mesh renderings and per-vertex L1 error maps over an utterance on TFHP. Our method produces distinct articulatory configurations at salient phonemes, supported by low per-vertex L1 errors across the lips and lower-face region. This indicates that our articulation-aware modeling reconstructs phoneme-specific motion accurately throughout the utterance. In the mesh renderings, the upper-face region remains comparable to competing methods, indicating that our model captures articulatory motion without affecting non-articulatory regions.

4.5.2 Sequence-Level Visible Articulatory Dynamics

Beyond sampling articulation at discrete time points, we further visualize the entire utterance trajectory using slit-scan visualizations (x–t and y–t slices) in Fig. 5. Slit-scan enables direct comparison of these trajectories across methods and exposes the sharpness of articulatory transitions through the curve slope. For each frame, we average a narrow strip (5 pixels) centered at the horizontal slit (LMC–RMC) and the vertical slit (UL–LL) to obtain 1D profiles, which are stacked over time to form x–t and y–t maps capturing lip spreading and mouth opening, respectively. We visualize slit-scan trajectories for the utterance segment *adopted*. In (b), the LMC and RMC trajectories of our method show distinct rounding and spreading patterns with steeper slopes at articulatory transitions, indicating distinct horizontal articulatory motions that match the ground-truth pattern. In (c), the UL and LL trajectories of our method show distinct opening and closing patterns with steeper slopes during rapid transitions, indicating more distinct vertical articulatory motions than competing methods.

4.6 Analysis

4.6.1 Topology-aware Composition

Table 3: Ablation of TAC module.

SAM	TAC	FVE↓	LVE↓	LDTW↓
✓	✗	0.812	0.255	0.161
✓	✓	**0.771**	**0.235**	**0.132**

Table 3 shows the ablation of the TAC module on VOCASET. SAM predicts directional articulatory motions defined by speech production studies and reaches between UniTalker and MemoryTalker in Table 1. We further introduce TAC to compose the per-direction motions through spatial coupling across the facial surface, improving all metrics and yielding surface-consistent 3D facial animation. A comparison with a topology-independent composition baseline is provided in Appendix D.5.

4.6.2 Evaluation beyond Vertex-based Metrics

To verify that the improvement in visible articulation does not depend on vertex-based evaluation, we further evaluate it using a pretrained lip-reading expert, since all metrics above are computed on mesh vertices. Following the protocol of [40], we measure the Character Error Rate (CER) and Viseme Error Rate (VER) of the lip-reading expert on rendered animations. As shown in Table 4, our method achieves the lowest CER and VER among all compared methods. Since the lip-reading expert is trained independently of our method, these results confirm the improved visible articulation from an external perspective. Evaluation details are provided in Appendix E.5, together with an additional evaluation using electromagnetic articulography (EMA) measurements.

Table 4: Character Error Rate (CER) and Viseme Error Rate (VER) of a pretrained lip-reading expert on rendered VOCASET animations.

Methods	CER↓	VER↓
GT (reference)	41.70	34.50
FaceFormer	82.44	74.52
CodeTalker	<u>74.23</u>	<u>63.40</u>
Mimic	89.60	81.27
StreamingTalker	77.22	65.08
Ours	**71.98**	**62.78**

4.6.3 Directional Articulatory Modeling

Fig. 6 visualizes directional articulatory motion trajectories along three directions: Spreading–Rounding, Opening–Closing, and Protrusion–Retraction. Each trajectory is computed from lip-surface anchors (UL, LL, LMC, RMC), as detailed in Appendix Eq. (S3), and aligned with the speech waveform and per-phoneme mesh renderings. Baseline trajectories deviate from the ground truth in direction-specific ways. Some show flat Opening–Closing trajectories at opening-dominant phonemes (e.g., /aɪ/, /æ/, /aʊ/), others attenuated Protrusion–Retraction trajectories at protrusion-dominant phonemes (e.g., /tr/, /v/), and others noisy Spreading–Rounding trajectories at high-magnitude segments (e.g., /aʊ/). In contrast, our method closely follows the ground truth across all three directions, with consistent reconstruction quality in the mesh. (See Appendix F.4 for an additional comparison with all methods.)

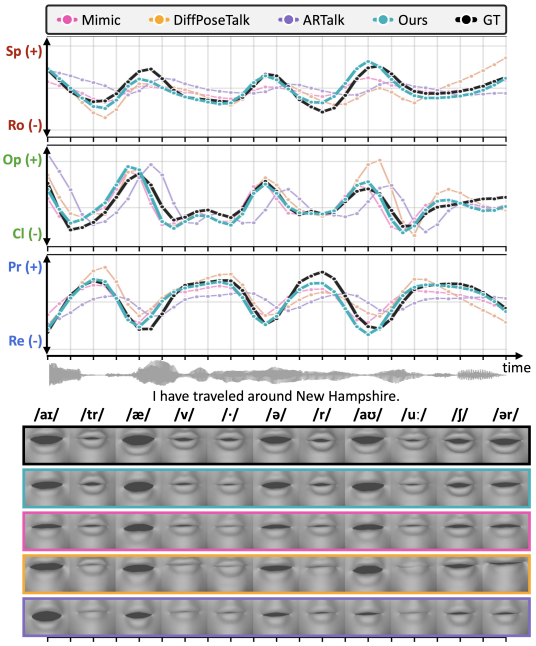


Figure 6: Analysis of Articulatory Trajectories.

4.6.4 Articulatory Transitions within Words

Fig. 7 visualizes the lip meshes and articulatory trajectories, extending the phoneme-level analysis of Fig. 6 to continuous articulatory changes within words. To this end, we select three words with dominant articulatory motions, “Strong”, “support”, and “nation”, and visualize consecutive lip meshes over each word. As shown in the figure, our method not only follows the overall trends of the ground truth but also closely tracks the continuous transitions between phonemes. In “Strong”, the lips, rounded for /ɹ/, spread toward /ɑŋ/, and in “support”, the mouth opens from the bilabial closure of /p/ toward the vowel /ɔ/. In “nation”, the lips open for the diphthong /eɪ/ and then protrude

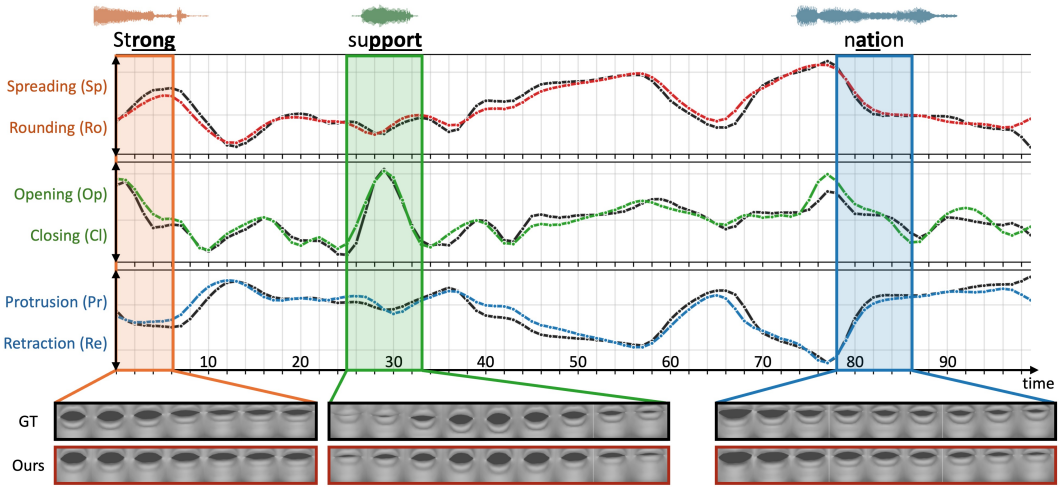


Figure 7: Directional articulatory motion trajectories of our method and the ground truth (Spreading–Rounding, Opening–Closing, Protrusion–Retraction) within words. Lip meshes are shown below for the highlighted words "Strong", "support", and "nation".

before the following postalveolar fricative /ʃ/, reflecting anticipatory lip protrusion [74]. These results suggest that, consistent with the ground truth, our method reproduces not only the articulation of individual phonemes but also the articulatory transitions between them.

4.7 User Study

Table 5 demonstrates the results of a pairwise A/B preference study following [48], comparing our method against seven competitors on **Lip Sync** and **Realism**. We recruit 26 participants and exclude 2 who failed attention checks, resulting in 24 valid responses. Each participant rates 15 randomly sampled clips per competitor, with two videos shown side-by-side in randomized order for forced-choice decisions on each criterion. Our method is preferred across all competitors, with average rates of 70.4% (Lip Sync) and 70.8% (Realism). A one-sided binomial test against chance level (50%) yields $p < 0.0001$ for every competitor, confirming statistical significance.

Table 5: User study: pairwise preference rates (%) on VOCASET.

Competitors	Lip Sync (% Ours)	Realism (% Ours)
vs. FaceFormer	71.4	65.8
vs. CodeTalker	72.2	75.0
vs. Imitator	77.8	73.1
vs. ScanTalk	73.1	79.7
vs. UniTalker	65.6	67.5
vs. MemoryTalker	63.9	68.6
vs. StreamingTalker	68.6	65.8

5 Conclusion

In this paper, we propose a novel articulation-aware approach to speech-driven 3D facial animation grounded in visible articulation during human speech production. We formulate speech-driven facial animation in terms of directional articulatory motions, together with Speech–Articulatory Memory (SAM) to model speech articulation correspondence and Topology-aware Articulatory Composition (TAC) to generate surface-consistent facial motion under mesh topology. Experimental results demonstrate that the proposed formulation improves the speech consistency of facial animation. By formulating facial motion through structured articulatory directions, our framework moves beyond holistic vertex regression toward articulation-grounded and interpretable speech-driven animation.

Acknowledgments and Disclosure of Funding

This research was supported by next-generation CultureTech technology development (R&D) project for cultural space AX transformation through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism in 2026. (Project Name : Development of experiential library technology that allows users to speak out and experience with their whole body. Project Number : RS-2026-25508598, Contribution Rate : 100%)

References

[1] Hiroki Tanaka and Satoshi Nakamura. The acceptability of virtual characters as social skills trainers: usability study. *JMIR human factors*, 9(1):e35358, 2022.

[2] Olov Engwall and Olle Bälter. Pronunciation feedback from real and virtual language teachers. *Computer Assisted Language Learning*, 20(3):235–262, 2007.

[3] Dominic W Massaro and Joanna Light. Using visible speech to train perception and production of speech for individuals with hearing loss. *Journal of speech, Language, and hearing research*, 47(2):304–320, 2004.

[4] Bishnu S Atal, Jih Jie Chang, Max V Mathews, and John W Tukey. Inversion of articulatory-to-acoustic transformation in the vocal tract by a computer-sorting technique. *The Journal of the Acoustical Society of America*, 63(5):1535–1555, 1978.

[5] Sankaran Panchapagesan and Abeer Alwan. A study of acoustic-to-articulatory inversion of speech by analysis-by-synthesis using chain matrices and the maeda articulatory model. *The Journal of the Acoustical Society of America*, 129(4):2144–2162, 2011.

[6] Prasanta Kumar Ghosh and Shrikanth Narayanan. A generalized smoothness criterion for acoustic-to-articulatory inversion. *The Journal of the Acoustical Society of America*, 128(4):2162–2172, 2010.

[7] Shinji Maeda. Compensatory articulation during speech: Evidence from the analysis and synthesis of vocal-tract shapes using an articulatory model. In *Speech production and speech modelling*, pages 131–149. Springer, Dordrecht, 1990.

[8] Benjamin Parrell, Vikram Ramanarayanan, Srikantan Nagarajan, and John Houde. Facts: A hierarchical task-based control model of speech incorporating sensory feedback. In *Interspeech 2018*, pages 1497–1501, 2018.

[9] 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. IEEE, 2019.

[10] Haozhe Wu, Songtao Zhou, Jia Jia, Junliang Xing, Qi Wen, and Xiang Wen. Speech-driven 3D face animation with composite and regional facial movements. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 6822–6830, New York, NY, USA, 2023. Association for Computing Machinery.

[11] Yingruo Fan, Zhaojiang Lin, Jun Saito, Wenping Wang, and Taku Komura. FaceFormer: Speech-driven 3D facial animation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18770–18780, New Orleans, LA, USA, 2022. IEEE.

[12] Jinbo Xing, Menghan Xia, Yuechen Zhang, Xiaodong Cun, Jue Wang, and Tien-Tsin Wong. CodeTalker: Speech-driven 3D facial animation with discrete motion prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12780–12790, 2023.

[13] Stefan Stan, Kazi Injamamul Haque, and Zerrin Yumak. FaceDiffuser: Speech-driven 3d facial animation synthesis using diffusion. In *ACM SIGGRAPH Conference on Motion, Interaction and Games (MIG '23), November 15–17, 2023, Rennes, France*, New York, NY, USA, 2023. ACM.

[14] Zhiyao Sun, Tian Lv, Sheng Ye, Matthieu Lin, Jenny Sheng, Yu-Hui Wen, Minjing Yu, and Yong-jin Liu. Diffposetalk: Speech-driven stylistic 3D facial animation and head pose generation via diffusion models. *ACM Transactions on Graphics (TOG)*, 43(4):1–9, 2024.

[15] Shivangi Aneja, Justus Thies, Angela Dai, and Matthias Niessner. Facetalk: Audio-driven motion diffusion for neural parametric head models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21263–21273, Seattle, WA, USA, 2024. IEEE.

[16] Peng Chen, Xiaobao Wei, Ming Lu, Hui Chen, and Feng Tian. DiffusionTalker: Efficient and compact speech-driven 3d talking head via personalizer-guided distillation. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, New York, NY, USA, 2025. IEEE.

[17] Yifan Yang, Zhi Cen, Sida Peng, Xiangwei Chen, Yifu Deng, Xinyu Zhu, Fan Jia, Xiaowei Zhou, and Hujun Bao. Streamingtalker: Audio-driven 3d facial animation with autoregressive diffusion model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 11766–11774, 2026.

[18] Hyung Kyu Kim, Sangmin Lee, and Hak Gu Kim. Analyzing visible articulatory movements in speech production for speech-driven 3D facial animation. In *2024 IEEE International Conference on Image Processing (ICIP)*, pages 3575–3579. IEEE, 2024.

[19] Hyung Kyu Kim and Hak Gu Kim. Learning Phonetic Context-Dependent Viseme for Enhancing Speech-Driven 3D Facial Animation. In *Interspeech 2025*, pages 3763–3767, 2025.

[20] Teja Rebernik, Jidde Jacobi, Roel Jonkers, Aude Noiray, and Martijn Wieling. A review of data collection practices using electromagnetic articulography. *Laboratory Phonology*, 12(1):6, 2021.

[21] International Phonetic Association. *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, Cambridge, UK, 1999.

[22] Akshay Anand, Chenxu Guo, Cheol Jun Cho, Jiachen Lian, and Gopala Anumanchipalli. Teaching machines to speak using articulatory control. *arXiv preprint arXiv:2510.05619*, 2025.

[23] Anders Löfqvist. Lip kinematics in long and short stop and fricative consonants. *The Journal of the Acoustical Society of America*, 117(2):858–878, 2005.

[24] Jorge C Lucero and Kevin G Munhall. Analysis of facial motion patterns during speech using a matrix factorization algorithm. *The Journal of the Acoustical Society of America*, 124(4):2283–2290, 2008.

[25] 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)*, pages 1153–1162. IEEE, 2021.

[26] Peiji Yang, Huawei Wei, Yicheng Zhong, and Zhisheng Wang. Semi-supervised speech-driven 3d facial animation via cross-modal encoding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21032–21041, 2023.

[27] Hao Li, Ju Dai, Xin Zhao, Feng Zhou, Junjun Pan, and Lei Li. Wav2Sem: Plug-and-play audio semantic decoupling for 3d speech-driven facial animation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 183–192, 2025.

[28] Balamurugan Thambiraja, Malte Prinzler, Sadegh Aliakbarian, Darren Cosker, and Justus Thies. 3diFACE: Synthesizing and editing holistic 3d facial animation. In *International Conference on 3D Vision 2025*, 2025.

[29] Shan He, Haonan He, Shuo Yang, Xiaoyan Wu, Pengcheng Xia, Bing Yin, Cong Liu, Lirong Dai, and Chang Xu. Speech4mesh: Speech-assisted monocular 3d facial reconstruction for speech-driven 3d facial animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14192–14202, 2023.

[30] Yixiang Zhuang, Baoping Cheng, Yao Cheng, Yuntao Jin, Renshuai Liu, Chengyang Li, Xuan Cheng, Jing Liao, and Juncong Lin. Learn2talk: 3d talking face learns from 2d talking face. *IEEE Transactions on Visualization and Computer Graphics*, 31(9):5829–5841, 2025.

[31] Aggelina Chatziagapi and Dimitris Samaras. Avface: Towards detailed audio-visual 4d face reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16878–16889, Vancouver, BC, Canada, 2023. IEEE.

[32] Kazi Injamamul Haque and Zerrin Yumak. FaceXHuBERT: Text-less speech-driven E(X)pressive 3d facial animation synthesis using self-supervised speech representation learning. In *International Conference on Multimodal Interaction (ICMI '23)*, New York, NY, USA, 2023. ACM.

[33] Kazi Injamamul Haque, Sichun Wu, and Zerrin Yumak. ProbTalk3D-X: Prosody enhanced non-deterministic emotion controllable speech-driven 3d facial animation synthesis. *Computers & Graphics*, page 104358, 2025.

[34] Yixiang Zhuang, Chunshan Ma, Yao Cheng, Xuan Cheng, Jing Liao, and Juncong Lin. TalkingEyes: Pluralistic speech-driven 3D eye gaze animation. *IEEE Transactions on Visualization and Computer Graphics*, 2026.

[35] Haoyu Zhang, Zhipeng Li, Yiwen Guo, and Tianshu Yu. Ex-Omni: Enabling 3d facial animation generation for omni-modal large language models. *arXiv preprint arXiv:2602.07106*, 2026.

[36] Sichun Wu, Kazi Injamamul Haque, and Zerrin Yumak. Probtalk3d: Non-deterministic emotion controllable speech-driven 3d facial animation synthesis using vq-vae. In *Proceedings of the 17th ACM SIGGRAPH conference on motion, interaction, and games*, pages 1–12, 2024.

[37] Karren D Yang, Anurag Ranjan, Jen-Hao Rick Chang, Raviteja Vemulapalli, and Oncel Tuzel. Probabilistic speech-driven 3d facial motion synthesis: New benchmarks methods and applications. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 27294–27303, 2024.

[38] Zhihao Xu, Shengjie Gong, Jiapeng Tang, Lingyu Liang, Yining Huang, Haojie Li, and Shuangping Huang. Kmtalk: Speech-driven 3d facial animation with key motion embedding. In *European Conference on Computer Vision*, pages 236–253. Springer, 2024.

[39] Ziqiao Peng et al. SelfTalk: A self-supervised commutative training diagram to comprehend 3D talking faces. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 5292–5301, New York, NY, USA, 2023. ACM.

[40] Han EunGi, Oh Hyun-Bin, Kim Sung-Bin, Corentin Nivelet Etcheberry, Suekyeong Nam, Janghoon Ju, and Tae-Hyun Oh. Enhancing speech-driven 3d facial animation with audio-visual guidance from lip reading expert. In *Proceedings of Interspeech 2024*, pages 2940–2944, Kos, Greece, 2024. ISCA.

[41] Xiangyu Fan, Jiaqi Li, Zhiqian Lin, Weiye Xiao, and Lei Yang. Unitalker: Scaling up audio-driven 3d facial animation through a unified model. In *European Conference on Computer Vision (ECCV)*, pages 204–221, Cham, 2024. Springer.

[42] Gabriele Fanelli, Juergen Gall, Harald Romsdorfer, Thibaut Weise, and Luc Van Gool. A 3-d audio-visual corpus of affective communication. *IEEE Transactions on Multimedia*, 12(6):591–598, 2010.

[43] Ziqiao Peng, Haoyu Wu, Zhenbo Song, Hao Xu, Xiangyu Zhu, Jun He, Hongyan Liu, and Zhaoxin Fan. EmoTalk: Speech-driven emotional disentanglement for 3D face animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20687–20697, October 2023.

[44] Cheng-hsin Wuu, Ningyuan Zheng, Scott Ardisson, Rohan Bali, Danielle Belko, Eric Brockmeyer, Lucas Evans, Timothy Godisart, Hyowon Ha, Xuhua Huang, et al. Multiface: A dataset for neural face rendering, 2022.

[45] Federico Nocentini, Thomas Besnier, Claudio Ferrari, Sylvain Arguillere, Stefano Berretti, and Mohamed Daoudi. Scantalk: 3d talking heads from unregistered scans. In *European Conference on Computer Vision*, pages 19–36. Springer, 2024.

[46] Xuangeng Chu, Nabarun Goswami, Ziteng Cui, Hanqin Wang, and Tatsuya Harada. ARTalk: Speech-driven 3d head animation via autoregressive model. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, SA Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery.

[47] Balamurugan Thambiraja, Ikhsanul Habibie, Sadegh Aliakbarian, Darren Cosker, Christian Theobalt, and Justus Thies. Imitator: Personalized speech-driven 3D facial animation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 20621–20631, October 2023.

[48] Hyung Kyu Kim, Sangmin Lee, and Hak Gu Kim. MemoryTalker: Personalized speech-driven 3d facial animation via audio-guided stylization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11241–11251, 2025.

[49] Hui Fu, Zeqing Wang, Ke Gong, Keze Wang, Tianshui Chen, Haojie Li, Haifeng Zeng, and Wenxiong Kang. Mimic: Speaking style disentanglement for speech-driven 3D facial animation. In *The 38th Annual AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[50] Zhaojie Chu, Kailing Guo, Xiaofen Xing, Pengsheng Liu, Bolun Cai, and Xiangmin Xu. Dcptalk: Speech-driven 3D face animation with personalized facial dynamic coupling properties. *IEEE Transactions on Multimedia*, 27:4427–4440, 2025.

[51] Radek Daněček, Kiran Chhatre, Shashank Tripathi, Yandong Wen, Michael Black, and Timo Bolkart. Emotional speech-driven animation with content-emotion disentanglement. In *SIGGRAPH Asia 2023 Conference Papers*, pages 1–13, 2023.

[52] Jiajian Xie, Shengyu Zhang, Mengze Li, Zhou Zhao, Fei Wu, et al. EcoFace: Audio-visual emotional co-disentanglement speech-driven 3d talking face generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

[53] Yifang Pan, Karan Singh, and Luiz Gustavo Hafemann. Model see model do: Speech-driven facial animation with style control. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*, pages 1–10, 2025.

[54] Jisoo Kim et al. DEEPTalk: Dynamic emotion embedding for probabilistic speech-driven 3d face animation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4275–4283, Washington, DC, USA, 2025. AAAI Press.

[55] Yihong Lin, Liang Peng, Zhaoxin Fan, Xianjia Wu, Jianqiao Hu, Xiandong Li, Wenxiong Kang, and Songju Lei. EmoFace: Emotion-content disentangled speech-driven 3d talking face animation. *arXiv preprint arXiv:2408.11518*, 2024.

[56] Zhan Qu, Shengyu Zhang, Mengze Li, Zhuo Chen, Chengfei Lv, Zhou Zhao, and Fei Wu. Exptalk: Diverse emotional expression via adaptive disentanglement and refined alignment for speech-driven 3d facial animation. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pages 1811–1819, 2025.

[57] Kang Shen, Haifeng Xia, Guangxing Geng, Guangyue Geng, Siyu Xia, and Zhengming Ding. Deitalk: Speech-driven 3d facial animation with dynamic emotional intensity modeling. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 10506–10514, 2024.

[58] Diqiong Jiang, Kai Zhu, Dan Song, Jian Chang, Chenglizhao Chen, and Zhenyu Wu. EditEmoTalk: Controllable speech-driven 3d facial animation with continuous expression editing. *arXiv preprint arXiv:2601.10000*, 2026.

[59] Xin Lu, Chuanqing Zhuang, Chenxi Jin, Zhengda Lu, Yiqun Wang, Wu Liu, and Jun Xiao. LSF-Animation: Label-free speech-driven facial animation via implicit feature representation. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, SA Conference Papers '25, New York, NY, USA, 2025. Association for Computing Machinery.

[60] Zhaojie Chu, Yilin Lan, Jianxiu Jin, Xiangmin Xu, and Xiaofen Xing. SiaTalker: Siamese emotion injection for coarse-to-fine speech-driven 3D facial animation. *IEEE Transactions on Affective Computing*, 2026.

[61] Zhaojie Chu, Kailing Guo, Xiaofen Xing, Bolun Cai, Lin Wang, and Xiangmin Xu. EETalk: Expression enhancement in speech-driven 3D facial animation. *IEEE Transactions on Multimedia*, 2026.

[62] Josh Chartier, Gopala K. Anumanchipalli, Keith Johnson, and Edward F. Chang. Encoding of articulatory kinematic trajectories in human speech sensorimotor cortex. *Neuron*, 98(5):1042–1054, 2018.

[63] Jun Wang, Jordan R Green, Ashok Samal, and Yana Yunusova. Articulatory distinctiveness of vowels and consonants: a data-driven approach. *Journal of Speech, Language, and Hearing Research*, 56(5):1539–1551, 2013.

[64] Victoria Fromkin. Lip positions in american english vowels. *Language and speech*, 7(4):215–225, 1964.

[65] Allen A Montgomery and Pamela L Jackson. Physical characteristics of the lips underlying vowel lipreading performance. *The Journal of the Acoustical Society of America*, 73(6):2134–2144, 1983.

[66] Yifan Sun, Qinlong Huang, and Xihong Wu. Unsupervised acoustic-to-articulatory inversion with variable vocal tract anatomy. In *Interspeech 2022*, pages 4656–4660, 2022.

[67] Woo-Jin Chung and Hong-Goo Kang. Speaker-independent acoustic-to-articulatory inversion through multi-channel attention discriminator. In *Interspeech 2024*, pages 1540–1544, 2024.

[68] Matthew S Craig, Pascal Van Lieshout, and Willy Wong. A linear model of acoustic-to-facial mapping: Model parameters, data set size, and generalization across speakers. *The Journal of the Acoustical Society of America*, 124(5):3183–3190, 2008.

[69] Pif Edwards, Chris Landreth, Eugene Fiume, and Karan Singh. Jali: an animator-centric viseme model for expressive lip synchronization. *ACM Transactions on graphics (TOG)*, 35(4):1–11, 2016.

[70] Pierre Badin, Gerard Bailly, Lionel Reveret, Monica Baciu, Christoph Segebarth, and Christophe Savariaux. Three-dimensional linear articulatory modeling of tongue, lips and face, based on mri and video images. *Journal of Phonetics*, 30(3):533–553, 2002.

[71] Yang Zhou, Zhan Xu, Chris Landreth, Evangelos Kalogerakis, Subhransu Maji, and Karan Singh. VisemeNet: Audio-Driven Animator-Centric Speech Animation. *ACM Transactions on Graphics (ToG)*, 37(4):1–10, 2018.

[72] Paul Ekman and Wallace V. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*. Consulting Psychologists Press, Palo Alto, CA, 1978.

[73] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4d scans. *ACM Trans. Graph.*, 36(6), November 2017.

[74] Aude Noiray, Marie-Agnes Cathiard, Lucie Ménard, and Christian Abry. Test of the movement expansion model: Anticipatory vowel lip protrusion and constriction in french and english speakers. *The Journal of the Acoustical Society of America*, 129(1):340–349, 2011.

[75] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021.

[76] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[77] K R Prajwal, Rudrabha Mukhopadhyay, Vinay P. Namboodiri, and C.V. Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *Proceedings of the 28th ACM International Conference on Multimedia*, MM '20, page 484–492, New York, NY, USA, 2020. Association for Computing Machinery.

[78] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[79] 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.

[80] Lee Chae-Yeon, Oh Hyun-Bin, Han EunGi, Kim Sung-Bin, Suekyeong Nam, and Tae-Hyun Oh. Perceptually accurate 3d talking head generation: New definitions, speech-mesh representation, and evaluation metrics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21065–21074, 2025.

[81] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. *arXiv preprint arXiv:1410.3916*, 2014.

[82] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1400–1409, 2016.

[83] Dong Gong, Lingqiao Liu, Vuong Le, Budhaditya Saha, Moussa Reda Mansour, Svetha Venkatesh, and Anton van den Hengel. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[84] Hyunjong Park, Jongyoun Noh, and Bumsub Ham. Learning memory-guided normality for anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14372–14381, 2020.

[85] Sangmin Lee, Hak Gu Kim, Dae Hwi Choi, Hyung-Il Kim, and Yong Man Ro. Video prediction recalling long-term motion context via memory alignment learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[86] Minsu Kim, Joanna Hong, Se Jin Park, and Yong Man Ro. Multi-modality associative bridging through memory: Speech sound recollected from face video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 296–306, 2021.

[87] Minsu Kim, Jeong Hun Yeo, and Yong Man Ro. Distinguishing homophenes using multi-head visual-audio memory for lip reading. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 1174–1182, 2022.

[88] Sangmin Lee, Hyung-Il Kim, and Yong Man Ro. Weakly paired associative learning for sound and image representations via bimodal associative memory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10534–10543, 2022.

[89] Se Jin Park, Minsu Kim, Joanna Hong, Jeongsoo Choi, and Yong Man Ro. Synctalkface: Talking face generation with precise lip-syncing via audio-lip memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2062–2070, 2022.

[90] Shuai Tan, Bin Ji, and Ye Pan. Emmn: Emotional motion memory network for audio-driven emotional talking face generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22146–22156, 2023.

[91] Mark Tiede, Carol Y Espy-Wilson, Dolly Goldenberg, Vikramjit Mitra, Hosung Nam, and Ganesh Sivaraman. Quantifying kinematic aspects of reduction in a contrasting rate production task. *The Journal of the Acoustical Society of America*, 141(5_Supplement):3580–3580, 2017.

[92] Abraham Savitzky and Marcel JE Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical chemistry*, 36(8):1627–1639, 1964.

[93] Pingchuan Ma, Alexandros Haliassos, Adriana Fernandez-Lopez, Honglie Chen, Stavros Petridis, and Maja Pantic. Auto-avsr: Audio-visual speech recognition with automatic labels. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The Abstract and Sec. 1 explicitly state our contributions and scope. Specifically, we (1) propose a first articulation-aware framework for speech-driven 3D facial animation grounded in structured visible articulation; (2) introduce Speech–Articulatory Memory (SAM) to model speech-to-articulation correspondence via key-value memory retrieval and decoding under phonetic context; (3) introduce Topology-aware Articulatory Composition (TAC) to compose directional articulatory motions into coherent 3D facial motion under mesh topology; and (4) demonstrate the effectiveness of SAM and TAC through quantitative, qualitative, and user-study evaluations on VOCASET and TFHP, achieving state-of-the-art performance on standard reconstruction metrics and improved articulatory fidelity.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See Appendix H, which discusses two limitations: (1) the FLAME and BIWI mesh topologies do not include intra-oral structures (teeth, tongue), limiting realism for phonemes that depend on tongue placement; and (2) our formulation targets speech-driven articulatory motion and does not explicitly model upper-face dynamics such as blinking, which leaves a remaining FDD gap on TFHP relative to DiffPoseTalk's stochastic upper-face sampling.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.

- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: This is an empirical paper. It does not include theorems, lemmas, or formal proofs; all equations describe the proposed network architecture and training objectives.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Sec. 3 fully describes the SAM and TAC architectures. Sec. 4.1 and Appendix B.3 specify the datasets and train/val/test splits for VOCASET and TFHP. Sec. 4.2 reports all training details (Adam optimizer with learning rate 10^{-4}, batch size 1, two-stage schedule of 250+150 epochs, memory slots $N = 32$, half-window $\Omega = 7$, $D_{\mathrm{SAM}} = 128$, $D_{\mathrm{TAC}} = 64$, single RTX 4090). Sec. 4.3 defines the evaluation metrics, and Appendix B and C.3 provide additional implementation and evaluation details (motion autoencoder architecture, autoencoder training, and periodic positional encoding for SAM).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: All datasets used in this paper (VOCASET and TFHP for the main experiments, BIWI for supplementary topology-generalization analysis) are publicly available, and we follow the standard preprocessing and split protocols established in prior work [11, 12, 14]. Sec. 3, Sec. 4.2, and Appendix B disclose the full architecture, hyperparameters, and training schedule needed to re-implement the method. Code will be released upon acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 4.1 and Appendix B.3 specify dataset splits following standard protocols [11, 12, 14]. Sec. 4.2 reports the optimizer (Adam, lr 10^{-4}, batch size 1), the two-stage training schedule, and all architectural hyperparameters (N, Ω, D_{SAM}, D_{TAC}). Sec. 4.3 defines the four standard reconstruction metrics (FVE, LVE, FDD, LDTW) and the visible

articulatory distance and velocity errors used at test time. Appendix D provides ablation studies and a sensitivity analysis (Appendix D.7) that justify the chosen design and hyperparameter values.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: For the user study (Sec. 4.7), we report a one-sided binomial test against the 50% chance level on 24 valid participant responses (after excluding 2 failed attention checks from 26 recruits), each rating 15 randomly sampled clips per competitor; all seven pairwise comparisons yield $p < 0.0001$.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Sec. 4.2 reports that all experiments are conducted on a single NVIDIA RTX 4090 GPU (24 GB VRAM) with batch size 1, using a two-stage schedule of 250 epochs (Stage 1, SAM) and 150 epochs (Stage 2, TAC).

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.

- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses only publicly available datasets under their release terms, and the user study collects only pairwise preference responses without personally identifiable information. Potential misuse risks (e.g., deepfakes) are discussed in Appendix H.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Appendix H discusses both positive societal impacts (applications in virtual reality, education, accessibility for hearing-impaired individuals, and content production) and negative impacts (risk of misuse for generating deceptive content such as deepfakes), along with mitigation strategies including the development of complementary detection methods and responsible release practices.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The proposed model outputs 3D vertex motion sequences rather than photo-realistic content, and therefore does not fall into the high-risk category described in this question (e.g., pre-trained language models, image/video generators, or scraped datasets). Broader misuse considerations and recommended responsible release practices are discussed in Appendix H.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets (VOCASET [9], TFHP [14], HDTF [79], BIWI [42]), the FLAME mesh template [73], and the pre-trained HuBERT model [75] are cited with their original publications and used in accordance with their respective release terms for non-commercial research.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce new datasets or benchmarks. The code supporting the proposed method will be released upon acceptance with appropriate documentation.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: Sec. 4.7 reports the user study protocol (26 participants recruited, 24 valid responses after attention checks, 15 randomly sampled clips per competitor, pairwise side-by-side A/B preference task). Appendix G provides representative screenshots of the questionnaire interface. Participation was voluntary, with no monetary compensation provided.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [No]

    Justification: The study consists of a simple pairwise preference test. The participants only viewed short 3D talking-head video clips. It does not involve the collection of personal or sensitive information and poses minimal risk. IRB approval was not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs are not used as a component of the proposed method.

    Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

CONTENTS

1 Introduction **1**

2 Related Works **3**

2.1 Speech-Driven 3D Facial Animation . . . 3

2.2 Articulatory Modeling for Speech and Animation . . . 3

3 Proposed Method **3**

3.1 Speech–Articulatory Memory (SAM) . . . 4

3.2 Topology-aware Articulatory Composition (TAC) . . . 5

4 Experiments **6**

4.1 Datasets . . . 6

4.2 Implementation Details . . . 6

4.3 Evaluation Metrics . . . 6

4.4 Quantitative Results . . . 7

4.5 Qualitative Results . . . 8

4.6 Analysis . . . 9

4.7 User Study . . . 10

5 Conclusion **10**

A Extended Related Work **27**

A.1 Evaluation of Visible Articulation . . . 27

A.2 Memory Networks . . . 27

B Implementation Details **27**

B.1 Motion Autoencoder . . . 27

B.2 Periodic Positional Encoding . . . 28

B.3 Dataset . . . 28

B.4 Loss Weights . . . 28

B.5 Model Complexity . . . 28

C Evaluation Details **29**

C.1 Metric Scales and Units . . . 29

C.2 Visible Articulatory Anchors . . . 29

C.3 Articulatory Tendency . . . 29

D Ablation Studies **30**

D.1 Component-wise Analysis . . . 30

D.2 Directional Articulatory Decomposition . . . 31

D.3 Speech–Articulatory Memory . . . 32

D.4 Motion Autoencoder . . . 33

D.5 Topology-aware Articulatory Composition . . . 33

D.6 Training Strategy . . . 34

D.7 Hyperparameter Sensitivity 35

E Generalization and Robustness **36**

E.1 Topology Generalization on BIWI 36

E.2 Articulatory Evaluation on TFHP 36

E.3 Zero-shot Cross-dataset Transfer 36

E.4 Robustness to Noisy Audio 36

E.5 Evaluation Independent of Vertex-based Metrics 37

F Qualitative Analysis **38**

F.1 Stage-wise Effect of TAC 38

F.2 Vertex-wise Gate Visualization 38

F.3 Acoustic Bridge Memory Addressing 39

F.4 Articulatory Motion Trajectory 39

F.5 Comparisons with State-of-the-Art 40

G User Study Details **42**

H Limitations and Broader Impacts **42**

A Extended Related Work

A.1 Evaluation of Visible Articulation

Speech-driven 3D facial animation is commonly evaluated with vertex-based reconstruction metrics such as LVE and FDD [11, 12], together with lip synchronization measures based on temporal alignment [47]. Recent work evaluates the perceptual accuracy of lip motion with speech–mesh representations [80] or with a pretrained lip-reading model [40]. We complement these metrics with articulatory distance and velocity errors along the three directions (Sec. 4.4.2), and additionally report lip-reading errors (Sec. 4.6.2) and correlation with EMA measurements (Appendix E.5), which are independent of our formulation.

A.2 Memory Networks

Memory networks store representative patterns in explicit memory slots and retrieve them through addressing [81, 82]. They have been used for anomaly detection [83, 84], video prediction [85], and cross-modal association between audio and visual signals [86–88], including lip-synchronized talking face generation [89] and emotional talking faces [90]. In speech-driven 3D facial animation, MemoryTalker [48] uses memory for speaker-specific stylization. In contrast, SAM stores directional articulatory motion patterns in separate value memories and retrieves them through an acoustic key memory, so that the one-to-many mapping from speech to motion is resolved within each articulatory direction.

B Implementation Details

B.1 Motion Autoencoder

Architecture. The motion autoencoder used for $\mathcal{L}_{\text{store}}$ consists of three independent encoder–decoder pairs, each corresponding to one articulatory direction $\delta \in \{S, O, P\}$. Each pair operates on the corresponding ground-truth articulatory motion $\mathbf{v}_\delta^t \in \mathbb{R}^V$, encoding it into the latent motion feature $\mathbf{z}_\delta^t \in \mathbb{R}^{D_{\text{SAM}}}$ through the Transformer encoder and reconstructing $\mathbf{v}_\delta^t$ from $\mathbf{z}_\delta^t$ via the Transformer decoder. The directional latent motion feature $\mathbf{z}_\delta^t$ is used as the alignment target for the retrieved feature $\mathbf{f}_{\text{retri},\delta}^t$ in Eq. 5. The architecture for each direction is summarized in Table S1.

Table S1: Architecture of the motion autoencoder for each articulatory direction, used to produce the directional latent motion feature $\mathbf{z}_\delta^t$.

Stage	Component	Output shape
Input	$\mathbf{v}_\delta^{1:T}$ (directional motion)	(T, V)
Encoder	Input projection ($V \to D_{\text{SAM}}$, Linear)	(T, D_{SAM})
	Sinusoidal positional encoding	(T, D_{SAM})
	Transformer encoder (1 layer, 4 heads, FFN $4\times D_{\text{SAM}}$, GELU, dropout 0.1)	(T, D_{SAM})
Latent	$\mathbf{z}_\delta^{1:T}$ (motion feature)	(T, D_{SAM})
Decoder	Transformer decoder (1 layer, 4 heads, FFN $4\times D_{\text{SAM}}$, GELU, dropout 0.1)	(T, D_{SAM})
	Output projection ($D_{\text{SAM}} \to V$, Linear)	(T, V)

Training. Across both datasets, VOCASET [9] and TFHP [14], autoencoders corresponding to the three articulatory directions are first pre-trained and then frozen during SAM training. For each sequence, the directional articulatory motion $\mathbf{v}_\delta^{1:T}$ is normalized along the time axis using z-score statistics computed for each vertex, passed through the corresponding encoder–decoder pair, and then unnormalized back to the raw vertex domain. The training objective minimizes the sum of the reconstruction loss and the velocity loss in the raw vertex domain:

$$\mathcal{L}_{\text{AE}} = \sum_{\delta \in \{S,O,P\}} \left(\mathcal{L}_{\text{recon},\delta}^{\text{AE}} + \mathcal{L}_{\text{vel},\delta}^{\text{AE}} \right), \tag{S1}$$

where $\mathcal{L}^{\mathrm{AE}}_{\mathrm{recon},\delta}$ and $\mathcal{L}^{\mathrm{AE}}_{\mathrm{vel},\delta}$ are the mean squared error (MSE) between the predicted and ground-truth articulatory motion and the MSE between their first-order temporal differences, respectively. We train for up to 250 epochs with Adam (learning rate 10^{-4}, batch size 1) on a single NVIDIA RTX 4090, with D_{SAM}=128. After pre-training, the autoencoders for each articulatory direction are frozen, and the directional latent motion feature $\mathbf{z}^t_\delta$ from their encoders is used to supervise the directional value memory $\mathbf{M}_{\mathrm{val},\delta}$ via $\mathcal{L}_{\mathrm{store}}$.

B.2 Periodic Positional Encoding

Periodic positional encoding (PPE) [11] periodically provides temporal order information to the Transformer inputs. Unlike standard sinusoidal positional encoding, which assigns a unique embedding to every position, PPE repeats the set of positional embeddings every ρ time steps across the sequence. Specifically, $\mathbf{P} \in \mathbb{R}^{T\times D_{\mathrm{SAM}}}$ is defined for each time step $t \in \{1,\dots,T\}$ and dimension-pair index $d \in \{0,\dots,\frac{D_{\mathrm{SAM}}}{2}-1\}$ as:

$$\mathbf{P}_{t,2d} = \sin\!\left(\frac{t \bmod \rho}{10000^{2d/D_{\mathrm{SAM}}}}\right), \quad \mathbf{P}_{t,2d+1} = \cos\!\left(\frac{t \bmod \rho}{10000^{2d/D_{\mathrm{SAM}}}}\right). \tag{S2}$$

Here $t \bmod \rho$ denotes the remainder when t is divided by ρ (e.g., $26 \bmod 25 = 1$). We set ρ to ensure that the period corresponds to one second of speech, considering the frame rate of each dataset [11].

B.3 Dataset

VOCASET [9] consists of 480 sequences from 12 speakers in FLAME topology [73], with an 8/2/2 speaker split for train/val/test. VOCASET is originally recorded at 60 fps. Following the standard protocol in prior works [11, 12], we set the frame rate to 30 fps and use its 255 unique sentences for evaluation. This ensures that all methods are evaluated at the same temporal resolution across all experimental settings.

TFHP [14] contains 1,052 videos from 588 speakers, covering lectures, interviews, and news programs, provided at 25 fps. TFHP provides two variants: with and without head pose. Since our framework focuses on articulatory motion and does not model rigid head pose, we use the variant without head pose, which yields $\sim$2.38M frames. The data is split 460/64/64 by speakers for train/val/test. TFHP is represented using FLAME [73] parameters, including expression and jaw-pose parameters for each frame and shape parameters for each speaker. We use the TFHP vertex sequences decoded from the FLAME parameters into 5,023-vertex meshes using the official FLAME model, so that all meshes are expressed in the canonical FLAME coordinate frame without rigid head motion.

Audio is resampled to 16 kHz and converted to mel-spectrograms with a window size of 1024 samples, a hop length of 256 samples, and 128 mel bins. The encoded acoustic features are linearly interpolated along the time axis to align with the mesh frame rate of each dataset.

B.4 Loss Weights

We set $\lambda_1 = 10^{-6}$ to balance the scale of $\mathcal{L}_{\mathrm{store},\delta}$ with that of the directional reconstruction loss $\mathcal{L}_{\mathrm{rec},\delta}$, and we set $\lambda_2 = 10^3$ to balance the scale of $\mathcal{L}_{\mathrm{lap}}$ with that of the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$.

B.5 Model Complexity

Table S2 reports the number of learnable parameters and the inference cost of each module. The motion autoencoder is used only to provide the reference features for $\mathcal{L}_{\mathrm{store}}$ during training and is not used at inference. Excluding the pre-trained content encoder [75], which is shared with prior speech-driven animation methods [32], the proposed modules add 5.74M parameters in total. The full model processes one second of audio in 5.14 ms on a single NVIDIA RTX 4090.

Table S2: Number of parameters and inference time of each module on VOCASET. Inference time is measured per second of input audio.

Component	Params (M)	Inference time (ms / s of audio)
Content encoder (HuBERT-base, pre-trained)	94.37	1.40
Motion autoencoder (training only)	5.32	n/a
SAM	5.71	0.93
TAC	0.03	2.81
Total (inference)	100.11	5.14

C Evaluation Details

C.1 Metric Scales and Units

For clarity, we report the scales and units of all quantitative metrics used in the main paper and the appendix. Table S3 indicates the scale factors for each metric across datasets. The metrics on BIWI follow the scale factors reported in Table S15. The lip-reading metrics (CER and VER) in Sec. 4.6.2 are expressed in percent, and the Pearson correlation coefficient (PCC) in Appendix E.5 is reported without scaling.

Table S3: Scale factors and units for all reported metrics.

Metric	Unit	VOCASET	TFHP
FVE	m	$\times 10^{-3}$	$\times 10^{-2}$
LVE	m^2	$\times 10^{-4}$	$\times 10^{-3}$
FDD	m^2	$\times 10^{-8}$	$\times 10^{-8}$
LDTW	–	$\times 10^{-3}$	$\times 10^{-3}$

C.2 Visible Articulatory Anchors

The mesh-surface articulatory anchors are defined on each mesh topology [42, 73] as fixed sets of vertex indices corresponding to the upper lip (UL), lower lip (LL), left mouth corner (LMC), right mouth corner (RMC), and jaw. We manually select these vertex sets in Blender, following the lip and jaw sensor locations used in acoustic-to-articulatory inversion studies [20, 91]. For the FLAME topology, UL and LL each consist of 25 vertices, LMC and RMC each consist of 33 vertices, and the jaw consists of 17 vertices. Since VOCASET and TFHP share the FLAME topology, the same vertex indices are used for all speakers and sequences. The visible articulatory anchor position $\mathbf{p}_*(t)$ used in Eq. (10) is the mean position of the corresponding vertex set at frame t. The four lip anchors (UL, LL, LMC, RMC) are used for the articulatory metrics and trajectory analyses, while the jaw anchor is used only for the visualization in Figure 1.

C.3 Articulatory Tendency

We describe the detailed computation used for the articulatory tendency analysis shown in Figure 3. For each frame t, we compute three signed 1D trajectories from the four visible articulatory anchors (UL, LL, LMC, RMC).

$$\begin{aligned} s(t) &= (\mathbf{p}_{\mathrm{LMC},x}(t) - \mathbf{p}_{\mathrm{RMC},x}(t)) - (\bar{\mathbf{p}}_{\mathrm{LMC},x} - \bar{\mathbf{p}}_{\mathrm{RMC},x}), \\ o(t) &= (\mathbf{p}_{\mathrm{UL},y}(t) - \mathbf{p}_{\mathrm{LL},y}(t)) - (\bar{\mathbf{p}}_{\mathrm{UL},y} - \bar{\mathbf{p}}_{\mathrm{LL},y}), \\ p(t) &= (\mathbf{p}_{\mathrm{UL},z}(t) - \bar{\mathbf{p}}_{\mathrm{UL},z}) + (\mathbf{p}_{\mathrm{LL},z}(t) - \bar{\mathbf{p}}_{\mathrm{LL},z}), \end{aligned} \tag{S3}$$

where $s(t)$, $o(t)$, and $p(t)$ correspond to the spreading, opening, and protrusion signals. $\mathbf{p}_*(t)$ denotes the corresponding articulatory anchor coordinate at frame t, and $\bar{\mathbf{p}}_*$ denotes the temporal median of the corresponding signal across all frames. The trajectories ($s(t)$, $o(t)$, and $p(t)$) are centered by

their temporal medians to remove speaker-specific variations caused by differences in lip shape and size and to indicate the articulatory state relative to a sequence-specific reference. We also use the temporal median rather than the first frame to reduce sensitivity to the articulatory state of the first frame.

To determine whether the current motion is opening or closing, or protruding or retracting, we compute a local temporal slope centered at each frame using a first-order Savitzky–Golay derivative filter [92]:

$$\dot{y}(t) = \frac{\sum_{i=-K}^{K} i \cdot y(t+i)}{\sum_{i=-K}^{K} i^2}, \tag{S4}$$

where $y \in \{s, o, p\}$ and boundaries use reflective padding. We use a 7-frame window (K=3) centered at the current frame. This slope captures the local temporal trend over the $2K+1$ samples centered at t, providing a smoother estimate than a single-frame finite difference.

We classify each slope according to its sign and decompose it into the following directional components. Here, a positive slope corresponds to motion in the canonical direction (spreading, opening, protrusion), while a negative slope corresponds to its opposite (rounding, closing, retraction):

$$\begin{aligned} \mathrm{Sp}(t) &= \max(\dot{s}(t), 0), \quad \mathrm{Ro}(t) = \max(-\dot{s}(t), 0), \\ \mathrm{Op}(t) &= \max(\dot{o}(t), 0), \quad \mathrm{Cl}(t) = \max(-\dot{o}(t), 0), \\ \mathrm{Pr}(t) &= \max(\dot{p}(t), 0), \quad \mathrm{Re}(t) = \max(-\dot{p}(t), 0), \end{aligned} \tag{S5}$$

Here, Sp, Ro, Op, Cl, Pr, and Re denote spreading, rounding, opening, closing, protrusion, and retraction, respectively.

For each sequence $s \in \mathcal{S}$ with frame set $\mathcal{F}_s$, each tendency $k \in \{\mathrm{Sp}, \mathrm{Ro}, \mathrm{Op}, \mathrm{Cl}, \mathrm{Pr}, \mathrm{Re}\}$ is represented by $k(t)$ at each frame $t \in \mathcal{F}_s$.

$$S_k = \frac{1}{\sum_{s \in \mathcal{S}} |\mathcal{F}_s|} \sum_{s \in \mathcal{S}} \sum_{t \in \mathcal{F}_s} k(t), \tag{S6}$$

where S_k denotes the average magnitude of motion in direction k over all evaluation frames.

For each articulatory tendency k, $\hat{S}_k$ and S_k are computed from the generated and ground-truth motion sequences, respectively. We then compute their ratio:

$$r_k = \frac{\hat{S}_k}{S_k}, \tag{S7}$$

followed by the GT-normalized similarity:

$$\mathrm{sim}_k = \exp(-|\log r_k|), \tag{S8}$$

which equals 1 for an exact match ($r_k = 1$) and decreases symmetrically for overestimation ($r_k > 1$) or underestimation ($r_k < 1$) on a log scale. The radar plot in Figure 3 shows sim_k for each articulatory tendency (Spreading, Rounding, Opening, Closing, Protrusion, and Retraction).

D Ablation Studies

D.1 Component-wise Analysis

We report a staged comparison extending Table 3 of the main paper with a simpler articulation-aware baseline. The comparison separates how much of our performance comes from the directional articulatory decomposition itself and how much comes from the two proposed modules. As shown in Table S4, adding the directional articulatory decomposition to the plain baseline improves every metric, which confirms that articulation is a useful modeling axis. This configuration nevertheless remains well behind our full model, so the decomposition alone does not account for our results. The remaining gap is closed by the two proposed modules. SAM lowers the FVE from 0.836 to 0.812 by reducing the ambiguity in mapping a given acoustic input to directional articulatory motion, and TAC lowers it further to 0.771 by composing the directional articulatory motions over the mesh topology. Each stage improves on top of articulation, showing that our gains come from how articulation is modeled and composed rather than from the decomposition alone.

Table S4: Ablation on the directional articulatory decomposition, SAM, and TAC on VOCASET. "Baseline" denotes holistic vertex regression without any of the three components, and "Articulation only" applies the directional decomposition alone, without the memory banks in SAM and without TAC.

Method	Articulation	SAM	TAC	FVE↓	LVE↓	FDD↓	LDTW↓
Baseline	✗	✗	✗	0.844	0.291	0.154	0.188
Articulation only	✓	✗	✗	0.836	0.265	0.153	0.175
Articulation + SAM	✓	✓	✗	0.812	0.255	0.142	0.161
Ours	✓	✓	✓	**0.771**	**0.235**	**0.097**	**0.132**

D.2 Directional Articulatory Decomposition

To evaluate the effect of directional articulatory decomposition, we replace the three directional branches in SAM with a single branch that directly predicts holistic facial motion $\mathbf{v}^t \in \mathbb{R}^{V\times 3}$ without directional decomposition. For a parameter-matched comparison, we use the same number of parameters for both the unified and decomposed methods.

Figure S1 visualizes the rendered meshes and the displacement along each articulatory direction with and without directional decomposition. With directional decomposition, the model generates more distinct articulatory motion in each direction. This shows that decomposing and composing articulatory motion by direction is effective.

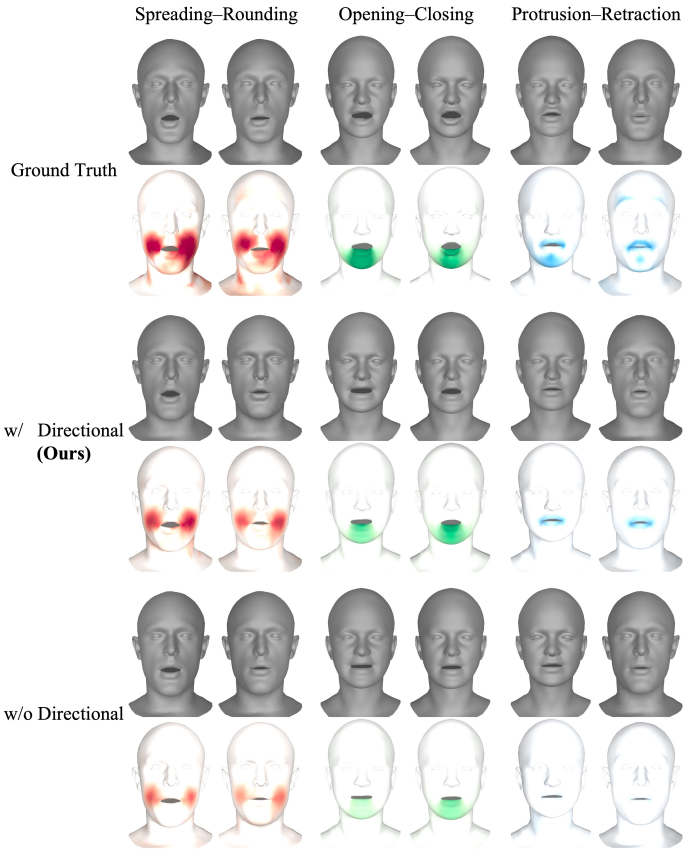


Figure S1: Rendered meshes and displacement heatmaps in the Spreading–Rounding (red), Opening–Closing (green), and Protrusion–Retraction (blue) directions with and without directional decomposition. Each column pair shows frames dominated by motion in the corresponding direction.

As shown in Table S5, the directional articulatory decomposition decreases FVE by 8.4%, LVE by 11.7%, FDD by 20.5%, and LDTW by 27.1% compared with the unified-branch baseline. This consistent improvement across all metrics demonstrates that decomposing facial motion into three directional articulatory motions (S/O/P) reduces cross-direction interference and improves overall facial dynamics.

Table S5: Ablation of directional articulatory decomposition on VOCASET.

Configuration	FVE↓	LVE↓	FDD ↓	LDTW ↓
w/o directional articulatory decomposition	0.842	0.266	0.122	0.181
w/ directional articulatory decomposition (S/O/P)	**0.771**	**0.235**	**0.097**	**0.132**

D.3 Speech–Articulatory Memory

Effect of Memory Structure. A key design choice in SAM is to store directional articulatory motion priors in explicit learnable memory banks using an acoustic bridge memory $\mathbf{M}_{\text{key}}$ and three directional articulatory value memories $\mathbf{M}_{\text{val},S}$, $\mathbf{M}_{\text{val},O}$, $\mathbf{M}_{\text{val},P}$. To evaluate the effect of this explicit storage, we remove $\mathbf{M}_{\text{key}}$ and $\mathbf{M}_{\text{val},\delta}$ while keeping the directional decomposition and TAC unchanged. Without these memory banks, the acoustic query $\mathbf{q}^t$ and content feature $\mathbf{f}^t_{\text{cont}}$ are fed directly to each directional articulatory motion decoder $g_\delta(\cdot,\cdot)$ to produce directional motion $\hat{\mathbf{v}}^t_\delta$. As shown in Table S6, our memory banks reduce FVE by 6.3%, LVE by 14.5%, FDD by 19.8%, and LDTW by 29.8% compared with the variant without memory banks. This result indicates that explicitly modeling the audio-to-motion mapping through learnable memory banks, rather than directly transforming audio into motion, contributes to improved performance.

Table S6: Ablation of the explicit motion memory banks in SAM on VOCASET.

Configuration	FVE↓	LVE↓	FDD↓	LDTW↓
w/o $\mathbf{M}_{\text{key}}, \mathbf{M}_{\text{val},\delta}$	0.823	0.275	0.121	0.188
Full (Ours)	**0.771**	**0.235**	**0.097**	**0.132**

Effect of Acoustic Query and Content Refinement. Our SAM module uses two speech representations within a directional retrieval pipeline. An acoustic query $\mathbf{q}^t$ from the Mel-spectrogram encoder addresses the acoustic bridge memory to retrieve the corresponding directional motion prior from each value memory. A content feature $\mathbf{f}^t_{\text{cont}}$ from HuBERT refines the retrieved motion via the Transformer decoder. To evaluate this design, we compare three query–refinement configurations while keeping the directional articulatory value memories and TAC module fixed. By changing the query and refinement features, we evaluate whether $\mathbf{f}^t_{\text{cont}}$ alone can be used for memory retrieval and whether $\mathbf{q}^t$ alone can be used for motion refinement, respectively.

As shown in Table S7, both substitutions degrade performance across all metrics, with the largest degradation observed in the Acoustic–Acoustic configuration, where content refinement is removed (i.e., without $\mathbf{f}^t_{\text{cont}}$). These results show that the acoustic feature $\mathbf{q}^t$ is more effective for memory retrieval, while the content feature $\mathbf{f}^t_{\text{cont}}$ is more effective for motion refinement.

Table S7: Ablation on the Query–Refinement design in SAM (VOCASET). "Acoustic" denotes Mel-spectrogram features and "Content" denotes HuBERT features.

Query	Refinement	FVE↓	LVE↓	FDD↓	LDTW↓
Content	Content	0.815	0.277	0.127	0.156
Acoustic	Acoustic	0.850	0.372	0.158	0.199
Acoustic	Content (Ours)	**0.771**	**0.235**	**0.097**	**0.132**

Effect of Value Memory Store Loss. The value memory store loss $\mathcal{L}_{\text{store},\delta}$ in Eq. (5) aligns the retrieved feature with the reference articulatory feature $\mathbf{z}_\delta$ for direction δ. To assess whether explicitly aligning the value memory with the reference feature $\mathbf{z}_\delta$ is effective, we remove $\mathcal{L}_{\text{store},\delta}$ from the

SAM objective. All other settings remain unchanged. Table S8 shows consistent degradation across all metrics when $\mathcal{L}_{\text{store}}$ is removed, indicating that explicitly aligning the value memory with $\mathbf{z}_\delta$ is effective.

Table S8: Ablation of the value memory store loss on VOCASET.

Configuration	FVE↓	LVE↓	FDD↓	LDTW↓
w/o $\mathcal{L}_{\text{store}}$	0.811	0.267	0.117	0.144
w/ $\mathcal{L}_{\text{store}}$ (Ours)	**0.771**	**0.235**	**0.097**	**0.132**

D.4 Motion Autoencoder

Effect of the Motion Autoencoder. To examine the contribution of the motion autoencoder, we replace the reference feature $\mathbf{z}_\delta^t$ used to align the value memory with a linear projection of the ground-truth directional motion $\mathbf{v}_\delta^t$ into the memory dimension D_{SAM}. Table S9 shows that this replacement consistently degrades all metrics. This indicates that the motion autoencoder provides a more effective reference feature for $\mathcal{L}_{\text{store}}$ than a linear projection.

Table S9: Ablation of the motion autoencoder on VOCASET. The autoencoder is replaced with a linear projection.

Reference feature	FVE↓	LVE↓	FDD↓	LDTW↓
Linear projection	0.811	0.257	0.159	0.150
Motion autoencoder (Ours)	**0.771**	**0.235**	**0.097**	**0.132**

Sensitivity to the Latent Dimension. To show the sensitivity to latent dimension in the motion autoencoder, we vary the latent dimension while keeping all other settings fixed. When the latent dimension differs from D_{SAM}, the latent feature of the motion autoencoder is linearly projected to D_{SAM} to obtain the reference feature $\mathbf{z}_\delta^t$. All SAM components, including the value memory, remain unchanged. As shown in Table S10, performance remains stable across the tested dimensions, with FVE varying only between 0.771 and 0.818. A latent dimension of 128 achieves the best results across all four metrics.

Table S10: Sensitivity to the latent dimension of the motion autoencoder on VOCASET.

Latent dimension	FVE↓	LVE↓	FDD↓	LDTW↓
32	0.783	0.238	0.131	0.145
64	0.779	0.236	0.105	0.142
128 (Ours)	**0.771**	**0.235**	**0.097**	**0.132**
256	0.802	0.240	0.117	0.141
384	0.809	0.245	0.112	0.148
512	0.818	0.252	0.116	0.134

Sensitivity to the Autoencoder Quality. To examine how the quality of the reference feature affects the final performance, we use motion autoencoder checkpoints from different training epochs, ranging from an undertrained checkpoint at epoch 50 to a converged checkpoint at epoch 250. We then train SAM with the reference feature obtained from each checkpoint. Table S11 shows that a lower autoencoder reconstruction error leads to consistently better animation accuracy, while the degradation with the undertrained autoencoder is gradual rather than abrupt.

D.5 Topology-aware Articulatory Composition

Effect of Topology-aware Composition. We ablate TAC by replacing it with a topology-independent MLP-based composition (Table S12). The MLP-based composition uses the same

Table S11: Sensitivity to the quality of the motion autoencoder on VOCASET. The reference feature is taken from autoencoder checkpoints at different training epochs. AE recon. error denotes the motion reconstruction MSE of the autoencoder.

AE checkpoint	AE recon. error ($\times 10^{-8}$)↓	FVE↓	LVE↓	FDD↓	LDTW↓
Epoch 50 (undertrained)	12.95	0.796	0.250	0.130	0.156
Epoch 150	8.39	0.789	0.241	0.122	0.150
Epoch 250 (Ours)	**7.89**	**0.771**	**0.235**	**0.097**	**0.132**

number of learnable parameters as TAC but processes each vertex in isolation, without modeling spatial coordination across mesh-connected vertices. In contrast, our GCN-based TAC couples motions across mesh-connected vertices via the facial topology and consistently outperforms the MLP composition on all metrics (FVE: 0.799 → 0.771, LVE: 0.243 → 0.235, FDD: 0.103 → 0.097, LDTW: 0.146 → 0.132). These results indicate that topology-aware composition is effective for integrating SAM's directional articulatory motions into surface-consistent 3D facial motion.

Table S12: Ablation of topology-aware composition on VOCASET. TAC is compared with a topology-independent MLP composition with the same number of learnable parameters.

Configuration	FVE↓	LVE↓	FDD↓	LDTW↓
SAM + MLP composition	0.799	0.243	0.103	0.146
SAM + TAC (Ours)	**0.771**	**0.235**	**0.097**	**0.132**

Effect of Graph Laplacian Consistency Loss. In addition to the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ and velocity loss $\mathcal{L}_{\mathrm{vel}}$ on the composed motion, we supervise TAC with a graph Laplacian consistency loss $\mathcal{L}_{\mathrm{lap}}$ that encourages the generated motion to match the ground truth in terms of local shape. For each vertex, $\mathbf{L}\tilde{\mathbf{v}}^t$ measures the difference between its motion and the normalized aggregation $\hat{\mathbf{F}}\tilde{\mathbf{v}}^t$ of the motions over its mesh neighborhood. $\mathcal{L}_{\mathrm{lap}}$ matches $\mathbf{L}\tilde{\mathbf{v}}^t$ to $\mathbf{L}\mathbf{v}^t$ of the ground truth, so that local shape relative to neighboring vertices on the mesh is also reproduced. While $\mathcal{L}_{\mathrm{rec}}$ penalizes the position error of each vertex, $\mathcal{L}_{\mathrm{lap}}$ penalizes errors in the local shape formed by neighboring vertices. To assess the effect of $\mathcal{L}_{\mathrm{lap}}$, we remove it from the TAC objective while keeping all other settings unchanged. As shown in Table S13, adding $\mathcal{L}_{\mathrm{lap}}$ improves all metrics, with the largest reductions in FDD (9.3%) and LDTW (5.0%).

Table S13: Ablation of the graph Laplacian consistency loss on VOCASET.

Configuration	FVE↓	LVE↓	FDD↓	LDTW↓
TAC w/o $\mathcal{L}_{\mathrm{lap}}$	0.788	0.236	0.107	0.139
TAC w/ $\mathcal{L}_{\mathrm{lap}}$ (Ours)	**0.771**	**0.235**	**0.097**	**0.132**

D.6 Training Strategy

Our training strategy first pretrains SAM in Stage 1 for 250 epochs and then freezes SAM while training TAC in Stage 2 for 150 epochs. To validate this strategy, we compare two alternatives. In *unfrozen fine-tuning*, SAM is initialized from the same Stage 1 checkpoint but left unfrozen, so SAM and TAC are optimized jointly in Stage 2. In *unified single-stage training*, SAM and TAC are trained jointly from scratch with the full objective. Table S14 shows that both alternatives underperform our frozen two-stage training strategy on all metrics. These results indicate that first learning directional articulatory motion with SAM and then learning surface-consistent composition with TAC while keeping SAM fixed is effective.

Table S14: Ablation of the training strategy on VOCASET.

Training Strategy	FVE↓	LVE↓	FDD↓	LDTW↓
Pretrained + unfrozen fine-tuning	0.798	0.244	0.122	0.144
Unified single-stage (from scratch)	0.801	0.245	0.118	0.142
Pretrained + frozen (Ours)	**0.771**	**0.235**	**0.097**	**0.132**

D.7 Hyperparameter Sensitivity

Effect of the Number of Memory Slots N. To determine the number of memory slots N and to examine whether the performance depends on the memory capacity, we vary $N \in \{16, 32, 64, 128\}$ on VOCASET while keeping all other hyperparameters fixed. Figure S2 shows that $N = 32$ achieves the lowest FVE and LVE, and we adopt this value in all experiments. Increasing the number of memory slots beyond $N = 32$ does not improve performance, indicating that the performance of SAM does not depend on increasing the memory capacity.

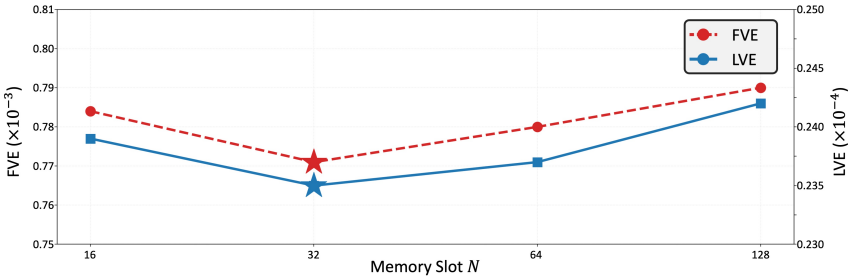


Figure S2: FVE and LVE on VOCASET with respect to the number of memory slots N. $N = 32$ ($\star$) achieves the lowest error on both metrics.

Effect of Temporal Window Size Ω. We investigate the effect of the half-window size Ω, which defines the temporal span of local phonetic context as a window of $2\Omega + 1$ frames. Figure S3 shows FVE and LVE on VOCASET as Ω varies over $\{0, 1, \ldots, 9\}$. Our method achieves the lowest FVE and LVE at Ω=7, while single-frame retrieval (Ω=0), in which the query and each memory slot cover only the current frame, yields the highest error. Although co-articulation is not explicitly modeled in our framework, this result indicates that retrieving directional motion over a temporal window, which allows each prediction to depend on its neighboring phonetic context, benefits the generated motion. Based on this analysis, we set Ω=7 in all experiments.

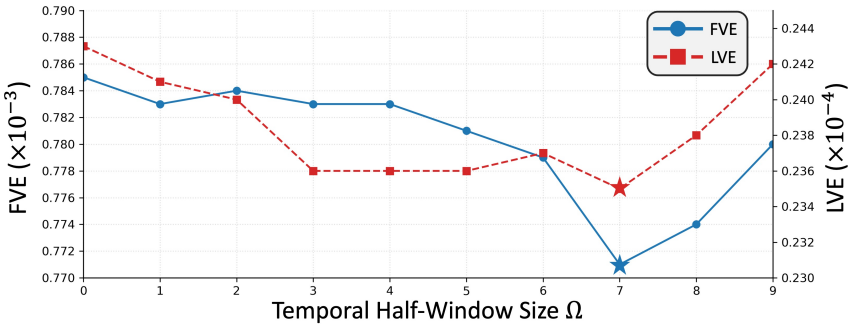


Figure S3: Effect of the half-window size Ω on FVE and LVE (VOCASET). Ω=0 corresponds to frame-independent retrieval without temporal motion context.

E Generalization and Robustness

E.1 Topology Generalization on BIWI

To validate that our method generalizes beyond FLAME topology, we evaluate on BIWI [42], whose mesh topology (23,370 vertices) differs substantially from FLAME (5,023 vertices). BIWI comprises 14 speakers each uttering 40 sentences twice (neutral and emotional) at 25 fps, with an average sequence length of 4.67 seconds. Following prior work [11, 12], we adopt the BIWI-Train (192 sentences, 6 subjects), -Val (24 sentences, 6 subjects), -Test-A (24 sentences, 6 subjects), and -Test-B (32 sentences, 8 subjects) splits. We train SAM+TAC with the same optimizer, learning rate, batch size, and training strategy as on VOCASET and TFHP. For BIWI, we set $N = 64$ and $D_{\mathrm{SAM}} = 256$ instead of $N = 32$ and $D_{\mathrm{SAM}} = 128$ to account for its denser mesh, and keep $D_{\mathrm{TAC}} = 64$. As shown in Table S15, our full model achieves the best FVE, LVE, and LDTW on BIWI.

Table S15: Topology generalization on BIWI (23,370 vertices).

Methods	FVE ($\times 10^{-1}$)↓	LVE ($\times 10^{-2}$)↓	FDD ($\times 10^{-5}$)↓	LDTW ($\times 10^{-2}$)↓
CodeTalker	0.124	0.127	0.461	0.319
SelfTalk	0.120	0.114	**0.148**	0.285
UniTalker	<u>0.112</u>	**0.106**	0.353	<u>0.283</u>
Ours (SAM only)	0.115	0.113	0.557	0.296
Ours (SAM+TAC)	**0.110**	**0.106**	<u>0.351</u>	**0.275**

E.2 Articulatory Evaluation on TFHP

We additionally report visible articulatory distance and velocity errors on TFHP in Table S16. Our method achieves the lowest errors on all three signals (ICW, ILD, LP) for both distance and velocity. These results generalize our VOCASET findings to TFHP, which spans various speaking styles.

Table S16: Visible articulatory distance and velocity errors on TFHP.

Methods	Distance Error			Velocity Error		
	ICW↓	ILD↓	LP↓	ICW↓	ILD↓	LP↓
Mimic	<u>4.320</u>	<u>5.326</u>	<u>5.968</u>	1.460	2.593	2.871
DiffPoseTalk	5.483	5.860	7.006	<u>1.221</u>	<u>2.168</u>	<u>2.438</u>
ARTalk	5.788	6.049	7.378	1.501	2.801	2.644
Ours	**4.299**	**4.829**	**5.859**	**1.168**	**2.083**	**2.414**

E.3 Zero-shot Cross-dataset Transfer

To evaluate generalization to an unseen dataset with different speakers and speaking conditions, we apply models trained on VOCASET to TFHP without any retraining or fine-tuning. Since all models are trained on VOCASET at 30 fps, we first generate motion at 30 fps and then downsample to 25 fps to match frame rate of the TFHP. For FaceFormer, CodeTalker, and StreamingTalker, which are conditioned on one-hot speaker identity, we report the best-performing result among the eight training speaker identities. For Mimic, which is conditioned on a style reference sequence, we report the result of the sequence that yields the best performance among the four speaker sequences provided in its official code. As shown in Table S17, our method achieves the lowest error on all seven metrics, including the articulatory metrics (ICW, ILD, and LP). These results indicate that our method generalizes to an unseen dataset better than the compared methods without retraining.

E.4 Robustness to Noisy Audio

We evaluate robustness to noisy audio by adding white Gaussian noise to the input waveform at 20, 10, and 5 dB SNR. Both models are trained on clean speech and evaluated under the same

Table S17: Zero-shot cross-dataset transfer from VOCASET to TFHP. Models are trained only on VOCASET and evaluated on TFHP without retraining or fine-tuning.

Methods	FVE↓	LVE↓	FDD↓	LDTW↓	ICW↓	ILD↓	LP↓
FaceFormer	0.187	0.185	1.099	0.442	4.966	7.923	8.523
CodeTalker	0.189	0.183	0.844	0.437	4.939	7.999	8.687
Mimic	0.179	0.182	1.452	0.454	5.123	8.812	8.557
StreamingTalker	0.172	0.180	0.623	0.439	4.771	8.543	8.453
Ours	**0.169**	**0.174**	**0.547**	**0.421**	**4.414**	**7.869**	**8.237**

corruption protocol on VOCASET. As shown in Table S18, our method achieves lower errors than StreamingTalker across all SNR levels. Notably, our method at 5 dB SNR still achieves lower errors on all four metrics than StreamingTalker at 10 dB SNR.

Table S18: Robustness to additive white Gaussian noise on VOCASET. Both models are trained on clean speech and evaluated at different SNR levels.

Methods	SNR	LVE↓	ICW↓	ILD↓	LP↓
StreamingTalker	5 dB	0.425	1.881	5.442	3.429
	10 dB	0.372	1.873	4.958	3.426
	20 dB	0.320	1.802	4.408	3.294
	Clean	0.272	1.658	3.955	2.478
Ours	5 dB	**0.361**	**1.792**	**4.655**	**3.396**
	10 dB	**0.314**	**1.767**	**3.891**	**3.352**
	20 dB	**0.244**	**1.725**	**3.504**	**3.101**
	Clean	**0.235**	**1.568**	**3.407**	**2.346**

E.5 Evaluation Independent of Vertex-based Metrics

For the lip-reading evaluation in Sec. 4.6.2, we render each generated mesh sequence into a frontal video and feed it to the pretrained Auto-AVSR visual speech recognition model [93]. We evaluate on 39 of the 40 VOCASET test sequences, since one sequence lacks ground-truth meshes in the public release.

Correlation with EMA Measurements To examine whether the generated articulation is consistent with physically measured lip movements, we compare the generated motion with electromagnetic articulography (EMA) recordings from the HPRC dataset [91]. HPRC contains synchronized audio and 100 Hz EMA recordings of eight American English speakers reading 720 IEEE sentences, including sensors on the upper lip and lower lip. We use all 12,870 speech–EMA recordings, including the repeated normal-rate takes provided in HPRC. All methods are trained on VOCASET and applied to HPRC without any retraining or fine-tuning. We extract the lip opening and lip protrusion trajectories from the generated meshes and compute the Pearson correlation coefficient (PCC) with the corresponding EMA lip sensor trajectories. On the mesh, we average the vertices in the UL and LL articulatory anchor sets, while for EMA, we use the UL and LL sensor positions. Lip opening is defined as the vertical UL–LL distance, and lip protrusion as the sum of the anterior displacements of UL and LL, using the corresponding axes in each coordinate system (FLAME y and z; EMA Z and X). Since EMA is sampled at 100 Hz and the generated motion at 30 fps, we resample each EMA trajectory to 30 fps by linear interpolation. We compute the PCC for each utterance and report the Fisher-z averaged PCC over all utterances. Lip spreading is not evaluated because HPRC provides only a single mouth-corner sensor, which does not allow measuring the inter-commissural width. Table S19 shows positive correlations with the EMA trajectories, and our method achieves the highest PCC for both lip opening and lip protrusion. Since EMA records articulator movements with sensors attached to the lips, this comparison also provides an evaluation independent of vertex-based reconstruction metrics.

Table S19: Pearson correlation coefficient (PCC) between the generated articulatory trajectories and the EMA lip sensor trajectories on HPRC, using models trained on VOCASET without retraining.

Methods	Lip opening↑	Lip protrusion↑
FaceFormer	0.609	0.558
CodeTalker	0.638	<u>0.609</u>
Mimic	0.517	0.605
StreamingTalker	<u>0.639</u>	0.601
Ours	**0.712**	**0.686**

F Qualitative Analysis

F.1 Stage-wise Effect of TAC

Figure S4 compares Stage 1 (SAM only) and Stage 2 (SAM + TAC) with the ground truth on TFHP using vertex-wise L1 error maps. SAM only already produces phoneme-consistent lip shapes through directional articulatory motions, but the remaining errors are concentrated around the lower face (lips, chin, and jawline). Adding TAC in Stage 2 reduces these errors by composing the directional articulatory motions over the mesh topology, producing surface-coherent reconstructions that align more tightly with the ground truth.

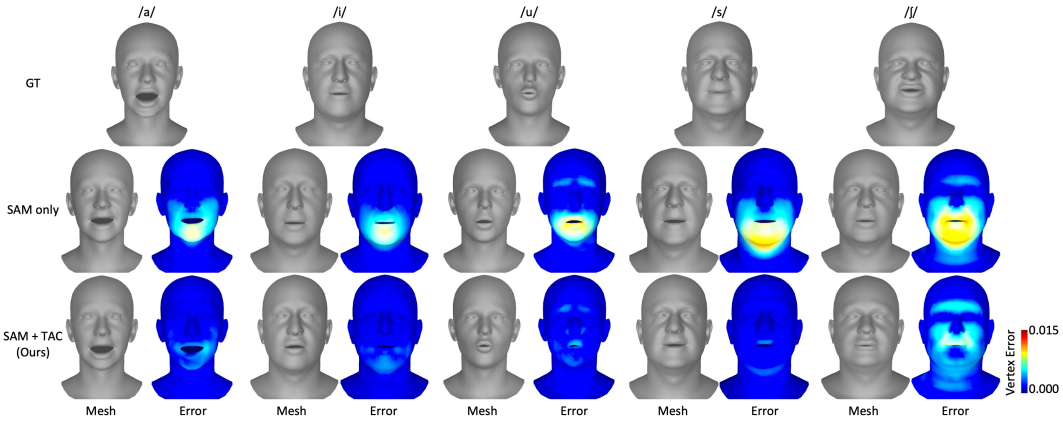


Figure S4: Stage-wise qualitative comparison on TFHP. Top row: GT. Middle row: Stage 1 (SAM only) result and vertex-wise error. Bottom row: Stage 2 (SAM + TAC) result and vertex-wise error.

F.2 Vertex-wise Gate Visualization

To examine whether the TAC modulation gate $\mathbf{m} = V \cdot \mathrm{softmax}(\tau_{\mathrm{TAC}}\boldsymbol{\alpha})$ forms an articulation-related spatial structure, we visualize the softmax distribution $\mathrm{softmax}(\tau_{\mathrm{TAC}}\boldsymbol{\alpha})$ on the FLAME template (Figure S5 (a)). We also aggregate the distribution by anatomical region (Figure S5 (b, c)). Since the softmax is taken over the V vertices, $\mathrm{softmax}(\tau_{\mathrm{TAC}}\boldsymbol{\alpha})$ sums to 1. Therefore, the uniform baseline is $1/V$, and any region whose mean exceeds $1/V$ receives a higher allocation than under a uniform distribution. The lip region receives the highest allocation at $2.0\times$ the uniform baseline of $1/V$, clearly exceeding all other regions. In contrast, the forehead, scalp, eye region, and ears stay near or below the baseline. This allocation emerges without any explicit supervision and is consistent with the region-dependent deformability that motivates the gate design, suggesting that the gate captures articulation-related spatial structure.

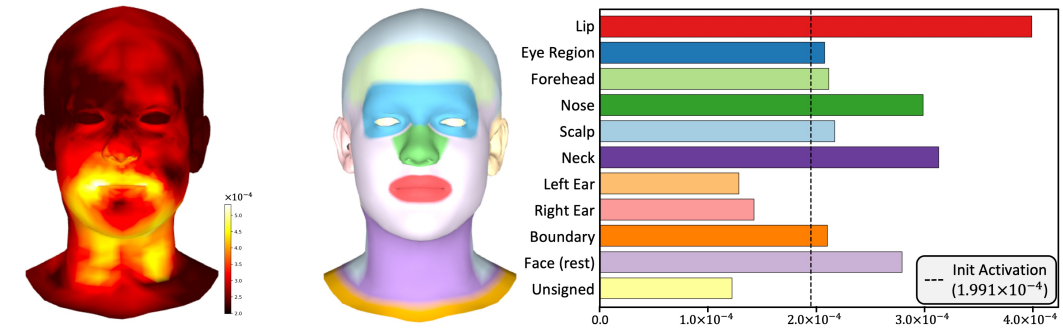


Figure S5: Visualization of the softmax distribution $\mathrm{softmax}(\tau_{\mathrm{TAC}}\boldsymbol{\alpha})$ underlying the TAC modulation gate $\mathbf{m}$ on the FLAME template. (a) Per-vertex distribution. (b) Anatomical region segmentation. (c) Region-grouped mean, with the dashed line indicating the uniform baseline $1/V$.

F.3 Acoustic Bridge Memory Addressing

Fig. S6 visualizes the addressing vector $\mathbf{A}^t$ over the $N{=}32$ slots of the acoustic bridge memory $\mathbf{M}_{\mathrm{key}}$ for two occurrences of four phoneme classes. The four classes cover distinct lip configurations: closed lips for /sil/, wide opening for /a:/, rounding and protrusion for /u/, and spreading for /i/.

Within each phoneme class, the addressing weights peak at the same slots in both occurrences, namely slots 2, 13, and 30 for /sil/, slots 13, 15, and 17 for /a:/, slots 14, 17, and 30 for /u/, and slots 2, 4, 18, and 29 for /i/. Each addressing vector concentrates on three or four of the 32 slots. Several slots are shared across classes (e.g., slot 2 by /sil/ and /i/, slot 17 by /a:/ and /u/, and slot 30 by /sil/ and /u/), so each phoneme class is identified by its combination of slots instead of a single slot. The relative weights still differ between occurrences, such as about 0.3 and 0.45 at slot 2 for /i/. Since the retrieved motion feature is a weighted sum of the value memory slots, these differences adjust the retrieved motion within the same phoneme class. One /a:/ occurrence also assigns a weight of about 0.2 to slot 30, which the other classes use for /sil/ and /u/. Because $\mathbf{A}^t$ is computed from the mel-spectrogram query and SAM is trained without phoneme labels, this phoneme-consistent addressing emerges without phoneme supervision.

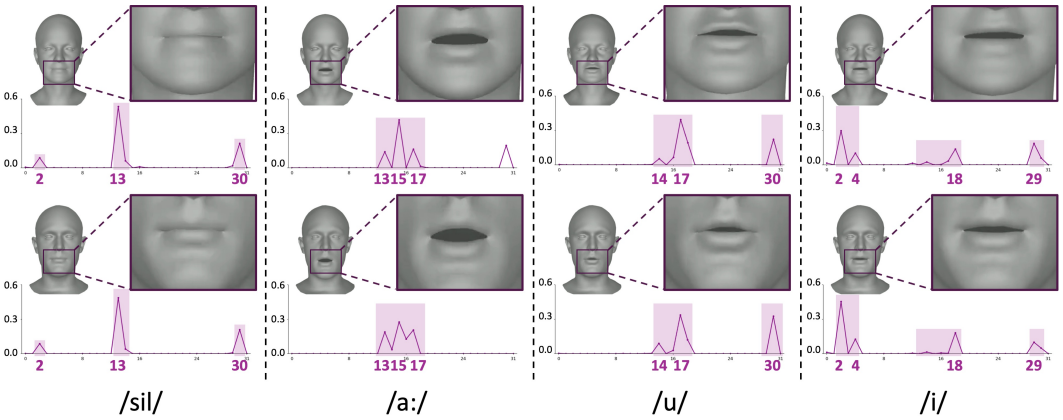


Figure S6: Addressing vectors of the acoustic bridge memory for each phoneme class across different occurrences.

F.4 Articulatory Motion Trajectory

Fig. S7 shows an additional example on a different utterance from Fig. 6 of the main paper, comparing our method with the baselines evaluated on TFHP (Mimic, DiffPoseTalk, and ARTalk). The trajectories are computed from the lip anchors (UL, LL, LMC, RMC) following Eq. (S3). Word-level transitions of our method are analyzed separately in Fig. 7 of the main paper.

The baseline deviations agree with the directional tendency strengths on TFHP in Fig. 3(b). ARTalk has the lowest Protrusion and Retraction strengths (0.47), and its Protrusion–Retraction trajectory stays nearly flat, without the retraction at /ɛ/–/i/ and /s/–/eɪ/. Its Opening–Closing trajectory also lags behind the ground truth from /aɪ/ to /v/. This lag is a timing error that the strength in Fig. 3 does not measure, since the strength compares only time-averaged magnitudes. Mimic underestimates the spreading peaks at /ɛ/–/i/ and /s/–/eɪ/, consistent with its Spreading and Rounding strengths of 0.56. DiffPoseTalk matches our strengths in Spreading, Rounding, Protrusion, and Retraction (0.89–0.93), but its Opening and Closing strengths drop to 0.81 and 0.80. In the trajectory, DiffPoseTalk underestimates the opening at /aɪ/ and opens the mouth at the bilabial /m/, where the ground-truth lips are closed. In contrast, our trajectories follow the ground truth in all three directions, with a slight overshoot at the /s/–/eɪ/ spreading peak.

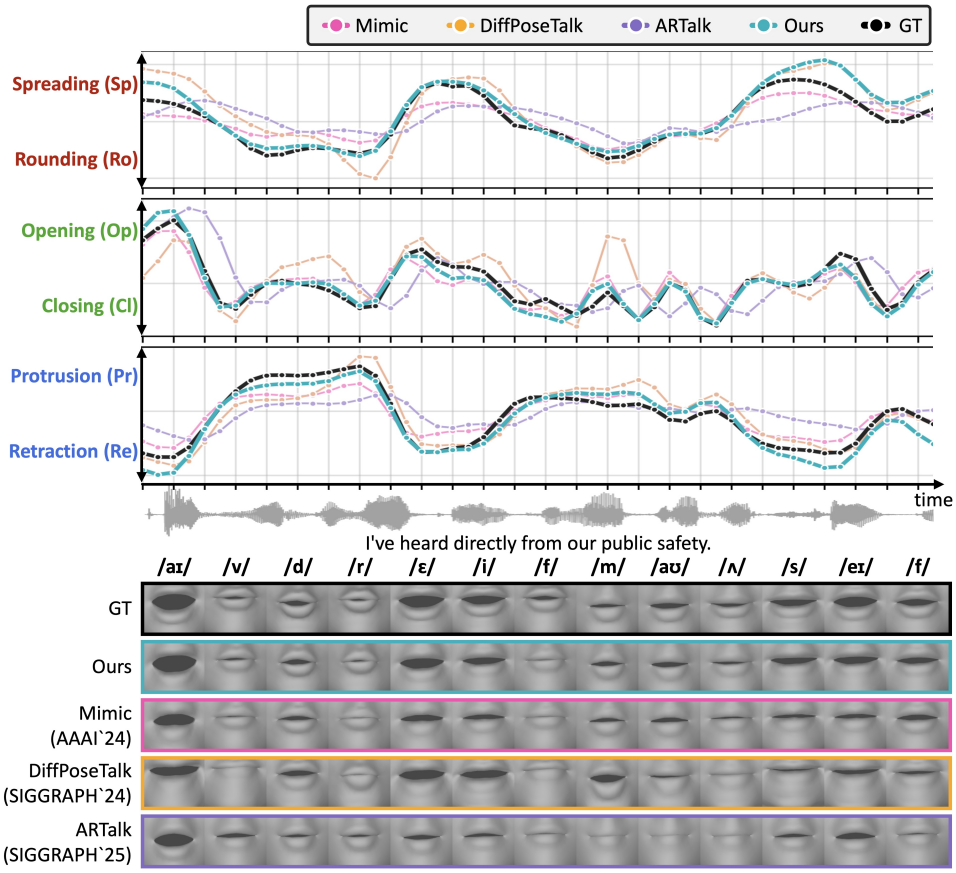


Figure S7: Additional example of directional articulatory motion trajectories (Spreading–Rounding, Opening–Closing, Protrusion–Retraction) of all compared methods and the ground truth for the utterance "I've heard directly from our public safety."

F.5 Comparisons with State-of-the-Art

Fig. S8 presents additional frame-level visual comparisons on VOCASET and TFHP. The first and third rows display full-face renderings, while the second and fourth rows show the per-vertex L1 error against the ground truth. Our method produces distinct articulatory configurations across phonemes, while prior methods tend to underestimate the magnitude of these motions. Across both datasets, our error maps remain low at the lips and lower-face region (jaw, chin), indicating that articulation-aware modeling improves articulatory reconstruction. The upper-face region remains comparable to competing methods, indicating that our model captures articulatory motion without affecting non-articulatory regions.

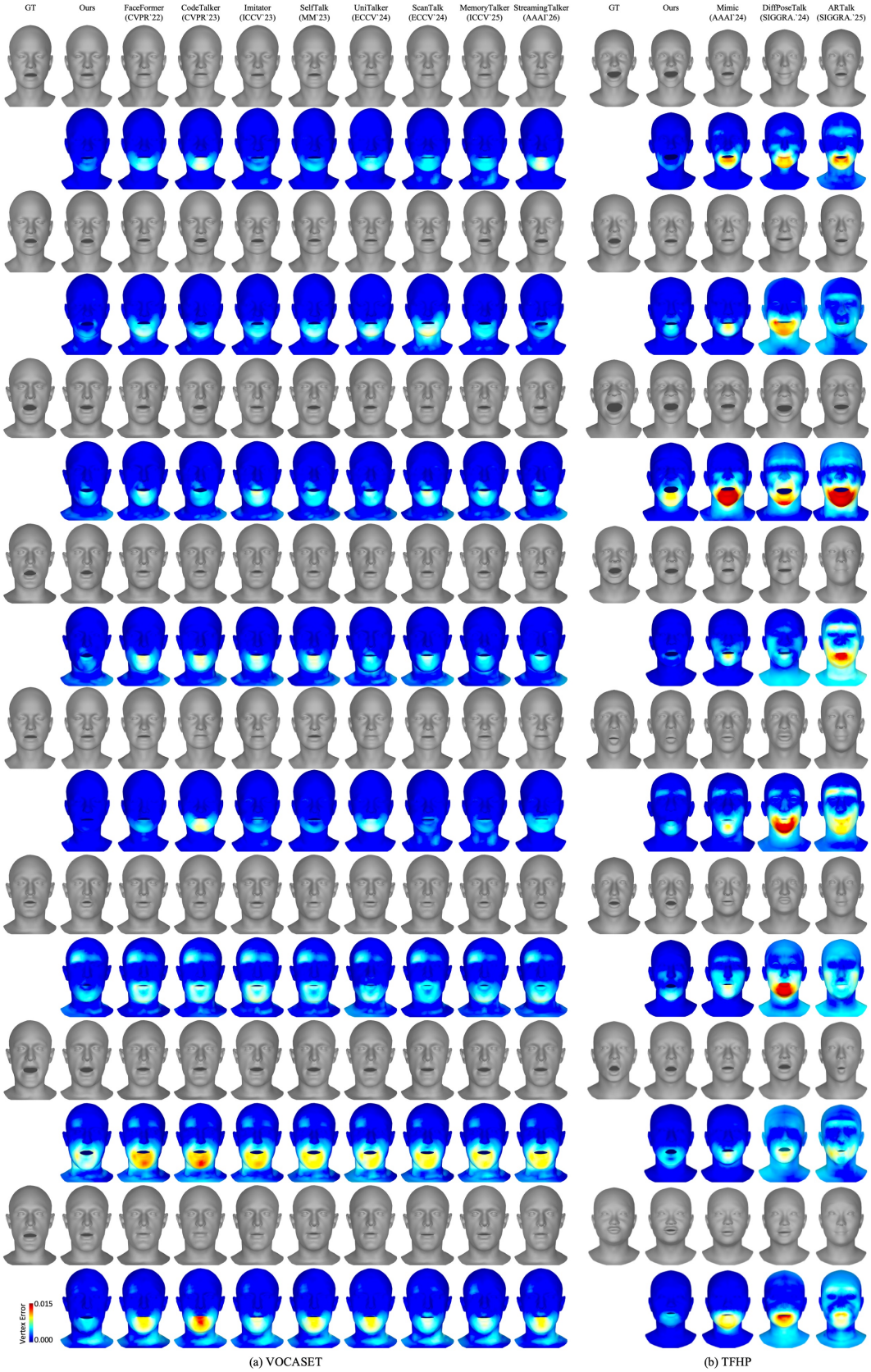


Figure S8: Additional qualitative comparisons on VOCASET and TFHP.

G User Study Details

We design our user study following the pairwise A/B preference protocol used in [48] for perceptual evaluation of speech-driven 3D facial animation.

Recruitment. We recruited 26 subjects through our extended network (e.g., colleagues and acquaintances). Two subjects who failed the attention checks were excluded, leaving 24 participants for analysis.

Compensation. Participation was voluntary, and no compensation was provided. This information was provided to all subjects before participation in the user study.

Informed Consent. The user study was conducted via a Google Forms questionnaire. Before participating in the user study, all subjects were informed about (i) the task: viewing side-by-side video clips of 3D facial animations and indicating their preference, (ii) the expected duration, (iii) the data being recorded (forced-choice responses only), (iv) the absence of compensation, and (v) the right to withdraw at any time. Only subjects who agreed to participate after reviewing the information proceeded to the questionnaire.

Privacy and Anonymization. We did not collect any personally identifiable information. The Google Forms questionnaire was configured to disable email collection and login requirements, so the researchers collecting the responses could not identify the subjects. Only forced-choice responses were recorded, and no response can be linked to a specific subject's identity.

Task and Instructions. Subjects viewed pairs of video clips of speech-driven 3D facial animations, presented side-by-side in randomized order. For each pair, subjects were asked two forced-choice questions: (1) which animation showed better lip synchronization with the audio, and (2) which animation appeared more realistic overall. Each subject rated 15 randomly sampled clips for each of the seven competitor methods. To identify inattentive subjects, we inserted 10 attention-check trials at random positions, in which one video was synchronized with the audio while the other was deliberately desynchronized. Subjects who failed to consistently identify the synchronized video were excluded. Representative interface screenshots are shown in Figure S9.

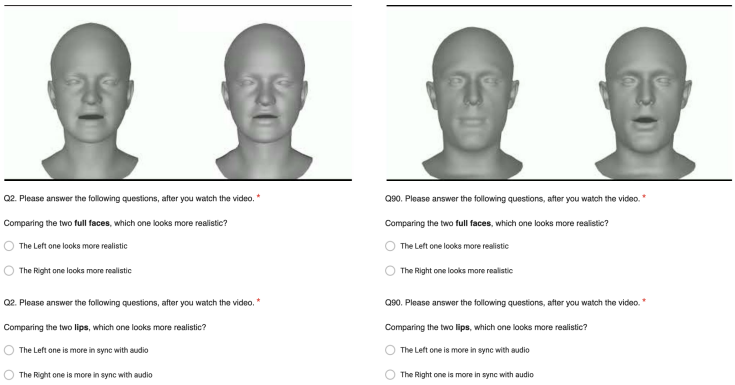


Figure S9: Examples of the conducted user study.

H Limitations and Broader Impacts

Limitations. Our method has two limitations that suggest directions for future work. First, we use the FLAME mesh topology (5,023 vertices) and BIWI mesh topology (23,370 vertices), neither of which includes intra-oral structures such as teeth and tongue. While the three directional articulatory motions control lip opening and thus influence when these structures become visible, explicitly modeling them would improve realism for phonemes that depend on tongue placement (e.g., /θ/). Second, our framework generates 3D facial motion from speech alone and does not explicitly model non-articulatory upper-face dynamics such as blinking. This limitation is reflected in FDD on

TFHP, where blinking is frequent. DiffPoseTalk reaches a lower FDD (0.321) by modeling such dynamics through diffusion-based stochastic sampling, while our audio-only method (0.578) attains an FDD comparable to ARTalk (0.581), which generates blinking autoregressively. Table S20 shows that methods without blinking modeling (Ours and Mimic) exhibit nearly identical eye-region FDD, whereas DiffPoseTalk and ARTalk achieve lower values through stochastic and autoregressive blinking generation respectively. Non-blinking regions (forehead, scalp) show minimal cross-method variation. On VOCASET, recorded under controlled conditions with limited upper-face activity, our method achieves the best FDD (0.097), confirming that accurate articulation does not come at the cost of upper-face degradation. Extending our framework with a stochastic blinking module remains future work.

Table S20: Region-wise FDD on TFHP ($\times 10^{-7}$ m^2, lower is better). Regions follow the official FLAME topology (Fig. S5 (b)).

Method	Forehead↓ (133 verts)	Eyes↓ (574 verts)	Scalp↓ (489 verts)
Mimic	0.020	0.099	0.010
DiffPoseTalk	**0.015**	**0.042**	**0.008**
ARTalk	0.044	0.060	0.011
Ours	0.023	0.102	0.010

Broader Impacts. Our framework advances speech-driven 3D facial animation with potential positive applications in virtual reality, education, accessibility for hearing-impaired individuals, and content production. However, improved facial animation quality also raises the risk of misuse for generating deceptive content such as deepfakes. We encourage the development of complementary detection methods and responsible release practices alongside deployment of such technologies.