

GaussianTalker: Speaker-specific Talking Head Synthesis via 3D Gaussian Splatting

Anonymous Authors

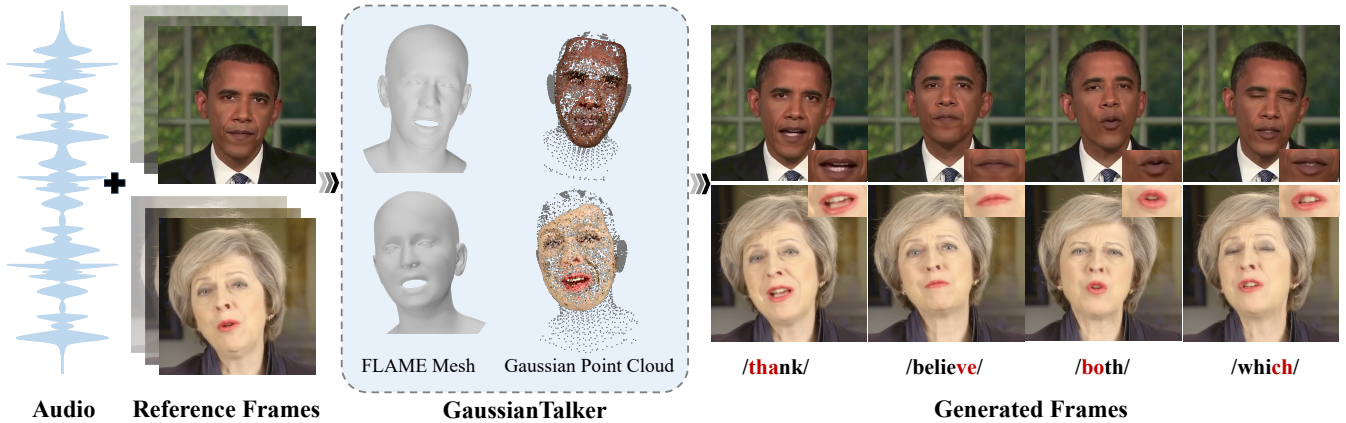


Figure 1: Given the driving audio and reference frames, GaussianTalker can animate the FLAME mesh of the speaker, which in turn drives the Gaussians associated with it, ultimately synthesizing a high-quality video with precise lip movements.

ABSTRACT

Recent works on audio-driven talking head synthesis using Neural Radiance Fields (NeRF) have achieved impressive results. However, due to inadequate pose and expression control caused by NeRF implicit representation, these methods still have some limitations, such as unsynchronized or unnatural lip movements, and visual jitter and artifacts. In this paper, we propose GaussianTalker, a novel method for audio-driven talking head synthesis based on 3D Gaussians. With the explicit representation property of 3D Gaussians, intuitive control of the facial motion is achieved by binding Gaussians to 3D facial models. GaussianTalker consists of two modules, Speaker-specific Motion Translator and Dynamic Gaussian Renderer. Speaker-specific Motion Translator achieves accurate lip movements specific to the target speaker through universalized audio feature extraction and customized lip motion generation. Dynamic Gaussian Renderer introduces Speaker-specific BlendShapes to enhance facial detail representation via a latent pose, delivering stable and realistic rendered videos. Extensive experimental results suggest that GaussianTalker outperforms existing state-of-the-art methods in talking head synthesis, delivering precise lip synchronization and exceptional visual quality. Our method achieves rendering speeds of 130 FPS on NVIDIA RTX4090 GPU, significantly

exceeding the threshold for real-time rendering performance, and can potentially be deployed on other hardware platforms.

CCS CONCEPTS

• Computing methodologies → Computer vision; Reconstruction;

KEYWORDS

talking head synthesis, 3D Gaussian splatting, speaker-specific

1 INTRODUCTION

Audio-driven talking head synthesis has attracted significant attention in various fields, including digital avatars, virtual reality, interactive entertainment, and online meetings[32, 44, 47]. This task aims to synthesize a video in which the target speaker’s lip movements are in sync with the given audio input.

In existing research, talking head synthesis approaches are primarily categorized into 2D-based and 3D-based approaches. Initially, most 2D-based approaches depended on Generative Adversarial Networks[16] or image-to-image translation[19] to synthesize talking heads in sync with audio. However, the absence of a unified facial model resulted in shortcomings in the identity preservation and pose control of the synthesized videos. More recently, research has been conducted to apply the Neural Radiance Fields (NeRF)[31] to this task. NeRF is a method that models continuous 3D scenes using implicit functions. Compared to 2D-based approaches, NeRF-based approaches can synthesize more lifelike talking head videos by effectively leveraging 3D facial modeling. Nevertheless, these approaches often encounter issues such as unsynchronized or unnatural lip movements, visual jitters, and sporadic artifacts, primarily because NeRF’s implicit definition entangles static facial geometry

Unpublished working draft. Not for distribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted by ACM, provided that the copies are not made for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

MM '24, October 28–November 01, 2024, Melbourne, Australia

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-XXXX-X/24/10

<https://doi.org/XXXXXXXXXXXXXX>

with dynamic motion, complicating the control over facial poses and expressions.

Recently, 3D Gaussian Splatting (3D GS)[23] has shown impressive achievements in 3D scene reconstruction. 3D GS employs 3D Gaussians as discrete geometric primitives, offering an explicit 3D scene representation and optimizing for real-time rendering performance. In comparison to NeRF, 3D GS not only significantly enhances rendering efficiency and visual quality, but also its paradigm based on 3D Gaussians is easier to control, making it potentially possible to flexibly and intuitively control facial movements. An intuitive strategy is to drive a Gaussian point cloud for facial motion by using parametric 3D facial models[6, 27, 34]. These models typically provide a comprehensive parameter space that allows for the control of facial attributes, such as shape, pose, and expression. By binding the Gaussians to the geometric topology of the model, dynamic talking heads can be generated by synchronizing the displacement of the bound Gaussians with changes in the facial attribute parameters.

Natural lip movements and realistic visual effects are crucial for talking head synthesis. However, the deviation of 3D facial models from a specific speaker’s face makes it challenging to synthesize lifelike videos using such a basic binding strategy. Specifically, there are *two main challenges*: 1) Distribution inconsistencies between the parameters driving the 3D Gaussians and the actual parameters can lead to deviations in point positions. This distribution inconsistency specifically refers to differences in speaking styles. For example, some individuals may speak with a small open mouth, while the driving signal is a wide open mouth. Lip movements that do not align with the speaker’s talking style present unnatural effects and even cause lip artifacts. 2) 3D facial models are limited in modeling complex faces. They can only capture macroscopic muscle movements but not fine details such as wrinkles and teeth. This inherent limitation hinders the 3D Gaussians’ accurate representation of a specific face, and the loss of facial details will also lead to visual flicker and artifacts.

For natural lip movements and realistic visual effects, bridging the subtle discrepancies between the 3D facial model and the specific speaker’s face is critical. Firstly, audio typically carries speaker identity information. It is necessary to decouple the driving audio from the original speaker’s identity information and ensure that the synthesized lip movements closely match the target speaker’s style. Secondly, intuitive control of Gaussian attributes helps break through the representation limitations of facial models. With a comprehensive set of blendshapes, these Gaussian attributes can be fine-tuned to capture facial details, refining the facial representation and further fitting a specific speaker.

In this work, we propose GaussianTalker, a framework designed to generate highly natural and realistic talking head videos, adaptable to multiple languages and various timbres of audio input. For dynamic reconstruction, 3D Gaussians are bound to the geometric topology of the parameterized FLAME[27], driving the Gaussians through the deformation of the FLAME mesh, ensuring accurate facial animation. To address the challenge of unnatural lip movements caused by inconsistent distributions, we propose a Speaker-specific Motion Translator. Employing timbre transformation-based contrastive learning for feature decoupling, a speaker-agnostic audio feature is extracted and merged with the target identity embedding

to generate facial poses and expressions that closely align with the target speaker. To confront the challenge posed by unrealistic visual effects caused by the inherent limitations of 3D facial models, the Dynamic Gaussian Renderer introduces Speaker-specific BlendShapes designed to predict the target speaker’s latent pose representation and refine facial details through the computation of detail-enhancing Gaussian attributes. It is important to note that our framework is not only capable of achieving speeds well beyond real-time rendering thresholds on contemporary GPUs, but it is also expected to be deployed on other hardware platforms, paving the way for its extensive application across multiple platforms.

The main contributions of this paper are summarized as follows:

- A novel framework for audio-driven talking head synthesis that utilizes 3D Gaussian Splatting bound to FLAME, which generates lifelike rendered videos by associating data from different modalities with specific speakers.
- A Speaker-specific Motion Translator decouples identity and employs personalized embedding for natural lip movements, while a Dynamic Gaussian Renderer refines Gaussian attributes through latent pose to ensure realistic visual effects.
- Extensive quantitative and qualitative experiments show GaussianTalker surpasses state-of-the-art methods in lip-sync and image quality, with its high rendering speed underscoring multi-hardware platform application potential.

2 RELATED WORK

2.1 Audio-Driven Facial Animation

Most early methods of audio-driven facial animation required establishing a mapping relationship between phonemes and visemes. These methods[3, 4] often overlook individual differences and are overly complex. The emergence of large-scale high-definition video datasets[1, 8, 12, 61], coupled with advancements in deep learning, has propelled the domain of learning-based audio-driven facial animation to the forefront of research. Initial efforts utilized 2D approaches[10, 14, 20, 36, 49, 52, 59, 63, 64], employing Generative Adversarial Networks[16] or image-to-image translation[19], to generate facial animations. However, these approaches fell short of accurately reproducing a speaker’s face due to lack of 3D facial modeling, leading to shortcomings in identity preservation and pose controllability. The adoption of 3D Morphable Models (3DMM)[2, 41, 66] has spurred research into audio-driven facial animation using 3DMM. These approaches[22, 30, 46, 51, 58], leveraging 3D modeling, are capable of rendering a more lifelike talking style compared to 2D approaches. Nevertheless, intermediate transformations through 3DMM may lead to information loss[2]. Moreover, prior universal 3D audio-driven animation methods[35, 39] generated vertex predictions with biases toward specific speakers, resulting in animations that fail to express personal styles.

Our method utilizes a parametric 3D facial model to represent the human face, generating facial motion representation from the driving audio. By identity decoupling and personalized embedding, we accommodate the unique styles of different speakers.

2.2 Human Face Rendering

Recent studies have adopted Neural Radiance Fields (NeRF)[31] for 3D human head modeling and rendering. NeRF encodes the

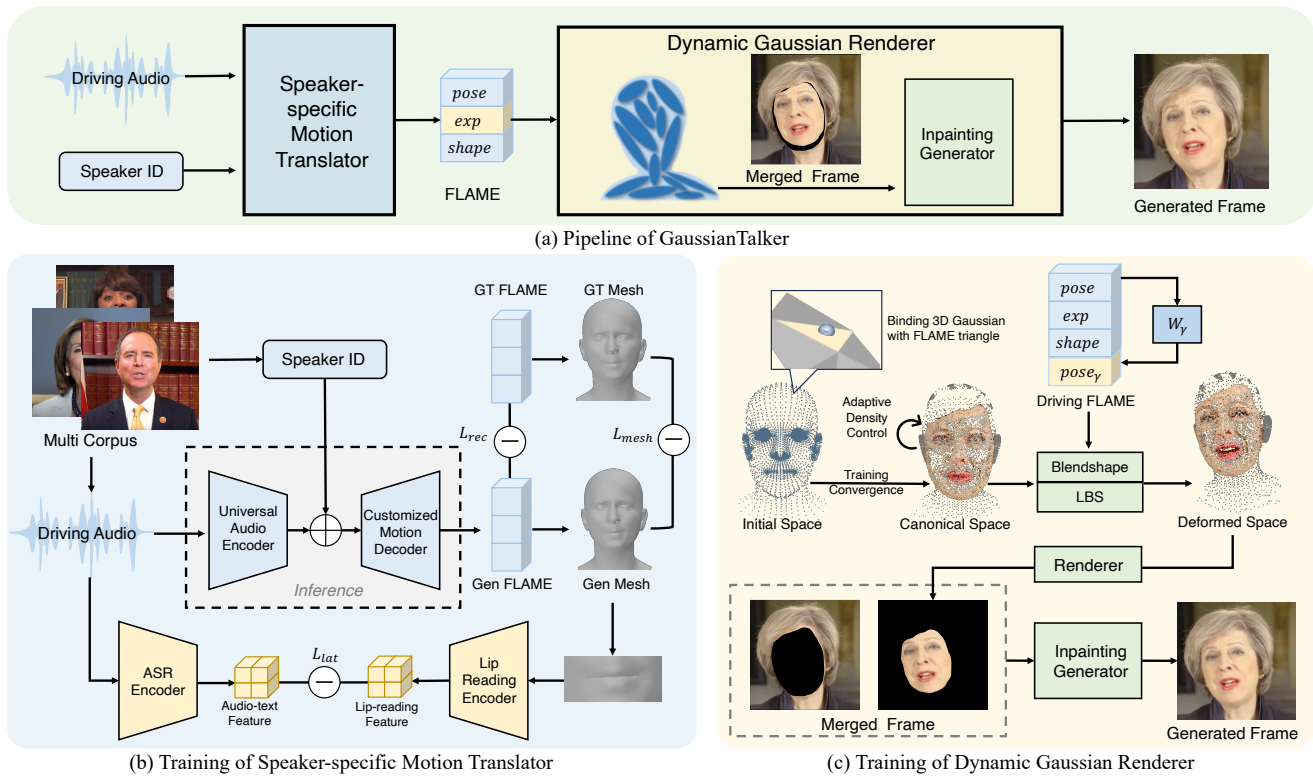


Figure 2: Overview of the proposed GaussianTalker. Subfigure (a) depicts speaker-specific FLAME generated from audio, driving Gaussians for rendering. Subfigure (b) illustrates the fusion of speaker-agnostic feature with speaker ID embeddings to decode FLAME. Subfigure (c) shows Gaussians driven by FLAME, subsequently rendering frames.

scene into a continuous volumetric field using a neural network, which enables high-quality 3D rendering. Following its successful application in dynamic scenes[15, 33, 37], NeRF has been naturally extended to talking head synthesis tasks[7, 17, 26, 29, 43, 45, 55–57]. Some studies have explored the implementation of talking face video generation in an end-to-end way[17, 29, 55]. Subsequent studies have realized several improvements in rendering efficiency[26, 45], few-shot synthesis[28, 43], and lip shape generalization[7, 56, 57].

With the advent of 3D Gaussian Splatting (3D GS)[23], considerable potential has been shown in enhancing both rendering efficiency and quality. Recent work[40, 42, 65] has implemented dynamic head reconstruction based on 3D GS. GaussianHead[50] employs learnable Gaussian diffusion for detailed head generation to accurately reproduce dynamic facial details. MonoGaussian-Avatar[11] utilizes 3D Gaussian representation and deformation fields for learning explicit avatars from monocular portrait videos.

Despite these approaches’ effectiveness, audio-driven talking head synthesis faces a significant challenge: cross-modal facial parameter generation from audio fails to precisely capture a specific person’s facial nuances. To our knowledge, we first apply 3D GS to this task and overcome the above problem. By Speaker-specific BlendShapes, our method facilitates the synthesis of more lifelike videos, charting a new course in the field of audio-driven high-fidelity talking head synthesis.

3 METHOD

3.1 Overview

In this section, we provide a detailed description of the architectural components of our proposed framework, GaussianTalker. GaussianTalker employs the parametric FLAME model[27] to serve as a bridge between facial animation and rendering. As shown in Figure 2, the overall framework consists of two main modules: 1) Speaker-Specific Motion Translator, which converts audio signals into speaker-specific FLAME parameters sequence for facial animation control; 2) Dynamic Gaussian Renderer, which utilizes FLAME to drive 3D Gaussians and renders dynamic talking head in real-time. The following subsections detail the design and training mechanisms of each module.

3.2 Speaker-specific Motion Translator

Given the diversity of audio inputs, we recognize that model generalization is challenging when solely dependent on video data from specific individuals. Consequently, we train the module using a multilingual, multi-individual dataset to improve its adaptability to diverse audio inputs. However, due to substantial variations in individual speaking styles, the distribution of FLAME parameters generated by the module might diverge from those of the target speaker, potentially compromising the rendered videos’ realism. To address this issue, we develop a Universal Audio Encoder that

decouples identity information from content information, and a Customized Motion Decoder that integrates personalization features. Additionally, we refer to SelfTalk[35] and introduce a lip-reading constraint mechanism based on self-supervision to further refine the synchronization of lip movements. This module’s detailed architecture and workflow are illustrated in Figure 2(b).

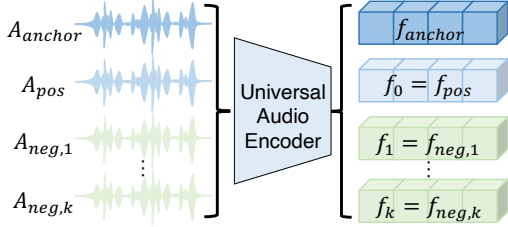


Figure 3: Negative audios $A_{neg,i}$ are obtained by dividing the audio and positive audio A_{pos} is obtained by timbre conversion. Audios are encoded to get the corresponding features for adversarial learning to fine-tune the encoder to become a Universal Audio Encoder.

3.2.1 Universal Audio Encoder. Audio signals often contain both speaker identity information and content information, which are usually tightly coupled. This indicates that in addition to conveying the content of speech, the audio also carries attributes unique to the speaker, such as pitch and timbre. In the task of audio feature extraction, our goal is to capture speech content while excluding information related to speaker identity. Therefore, we start at the audio encoding stage to achieve effective decoupling of identity and content information. We extract audio features using the audio encoder of a pre-trained Wav2Vec 2.0-based multilingual ASR model[54]. As shown in Figure 3, we segment a speaker’s audio into $k + 1$ segments, and then randomly select a segment as the anchor audio $A_{anchor} = A_{c_1,t_1}$. Next, we perform timbre conversion[38] on the anchor audio to obtain positive samples $A_{pos} = A_{c_1,t_2}$, while the remaining audio segments are used as negative samples. The subscripts c_i, t_i denote the content and timbre of the audio respectively. By encoding the audio, we get the respective feature, whose formulas are as follows:

$$f_t = E_{uni}(A_t; \theta_{uni}), \quad (1)$$

where f_t denotes the audio feature, E_{uni} and θ_{uni} denote the audio encoder and its learnable parameters, and A_t denotes the audio.

Positive samples are consistent in content with the anchor audio, and we anticipate them to generate similar audio features. For negative sample features, we aim for maximal dissimilarity from the anchor audio feature to achieve de-identification of audio encoding. To achieve this goal, we use the InfoNCE loss[48] for optimization. To maintain the original capability of the encoder, we use E_{asr} and D_{asr} , encoder and decoder of the pre-trained Wav2Vec 2.0 model[54] to guide the optimization:

$$\mathcal{L}_{adv} = -\log \frac{\exp(f_{anchor} \cdot f_0/\tau)}{\sum_{i=1}^k \exp(f_{anchor} \cdot f_i/\tau)}, \quad (2)$$

$$\mathcal{L}_{text} = \mathcal{L}_{ctc}(D_{asr}(f_{anchor}), D_{asr}(E_{asr}(A_{anchor}))), \quad (3)$$

where τ denotes the temperature.

3.2.2 Customized Motion Decoder. We introduce an identity embedding to generate motions that capture the speaker’s talking style, such as the difference in the degree of mouth opening when different speakers talk. The Identity embedding takes personal identity information as input and generates the personalized feature consistent with the audio feature dimensions. Next, the personalized feature is fused with the audio feature to serve as the input of the Customized Motion Decoder. The decoder based on a transformer-based structure, outputs the FLAME parameters sequence directly:

$$\hat{Y}_{1:T} = D_m(E_{id}(p; \theta_{id}) + E_{uni}(A), \theta_m), \quad (4)$$

where $\hat{Y}_{1:T} = (\hat{y}_1, \dots, \hat{y}_T)$ denotes the generated FLAME parameters sequence, D_m denotes the motion decoder, E_{id} denotes the identity embedding, p denotes the speaker’s identity, and θ_{id}, θ_m denote the learnable parameters of the corresponding component, respectively.

3.2.3 Training Object. The training objectives of Speaker-specific Motion Translator include three components: reconstruction, lip motion smoothness, and potential consistency.

Reconstruction. To ensure the generated motion sequence precisely aligns with the ground truth data, we calculate the mean square error of the ground truth $Y_{1:T} = (\mathbf{y}_1, \dots, \mathbf{y}_T)$ and the generated FLAME parameters sequence $\hat{Y}_{1:T}$, as well as the mean square error of the corresponding vertices of the FLAME mesh $V_{1:T} = (\mathbf{v}_1, \dots, \mathbf{v}_T)$ and $\hat{V}_{1:T} = (\hat{\mathbf{v}}_1, \dots, \hat{\mathbf{v}}_T)$:

$$\mathcal{L}_{rec} = \lambda_y \frac{1}{N \times T} \sum_{t=1}^T \|\mathbf{y}_t - \hat{\mathbf{y}}_t\|^2 + \lambda_v \frac{1}{K \times T} \sum_{t=1}^T \|\mathbf{v}_t - \hat{\mathbf{v}}_t\|^2, \quad (5)$$

where N denotes the dimension of FLAME parameters, K denotes the number of FLAME mesh vertices, T denotes the total number of video frames.

Lip motion smoothness. To avoid the potential motion jitter problem that results from relying solely on reconstruction loss, we enhance lip motion smoothness by calculating parameter changes across frames:

$$\mathcal{L}_{sth} = \lambda_{sth} \frac{1}{N \times T} \sum_{t=1}^T \|(\mathbf{y}_t - \mathbf{y}_{t-1}) - (\hat{\mathbf{y}}_t - \hat{\mathbf{y}}_{t-1})\|^2. \quad (6)$$

Potential consistency. During training, the generated motion sequence is passed to the lip-reading encoder to extract lip-reading feature. To ensure coherence between the audio-text feature and lip-reading feature, we compute their mean square error:

$$\mathcal{L}_{lat} = \lambda_{lat} \frac{1}{D \times T} \|E_{asr}(A) - E_{lip}(\hat{V}_{1:T}, M_L)\|^2, \quad (7)$$

where M_L denotes the index of lip region, E_{lip} denotes the lip encoder, and D denotes the feature dimension.

The final loss function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{sth} + \mathcal{L}_{lat}. \quad (8)$$

3.3 Dynamic Gaussian Renderer

3D GS[23] is a rasterization technique designed for the real-time rendering of photorealistic scenes. It utilizes a collection of 3D Gaussians for modeling, where each Gaussian in the model is characterized by the following learnable parameters: position $\mathbf{u} \in \mathbb{R}^3$, indicating the center of the Gaussian; rotation $\mathbf{r} \in \mathbb{R}^{3 \times 3}$, defining its orientation; scaling factor $\mathbf{s} \in \mathbb{R}^3$, controlling its size; opacity

$\alpha \in \mathbb{R}$, determining its visibility; and spherical harmonics coefficients $\kappa \in \mathbb{R}^{3 \times (k+1)^2}$, which determine the RGB color c through a k -order spherical harmonic function. This set of parameters for each Gaussian is denoted as $\mathcal{G} = \{\mathbf{u}, \mathbf{r}, \mathbf{s}, \alpha, \kappa\}$. During the rendering process, the color C of each pixel is determined by aggregating the contributions of all overlapping Gaussians, as described by the following equation:

$$C = \sum_{i \in N} c_i \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (9)$$

We aim to employ explicit 3D Gaussians to represent facial expressions, but static 3D Gaussians fall short of capturing dynamic changes in expressions. Consequently, we devised a Dynamic Gaussian Renderer that anchors the Gaussians to the FLAME triangles. This approach leverages FLAME’s BlendShapes and Skin Weights to control the deformation of the Gaussians. We also incorporate some Speaker-specific Blendshapes to enhance geometric and textural detail in facial rendering and introduce an Inpainting Generator to seamlessly integrate the rendered face with the original image. The complete pipeline is presented in Figure 2(c).

3.3.1 Dynamic Deformation. Given a set of FLAME parameters, β for shape, ϵ for expression, and ψ for pose changes, the vertices of the FLAME model in the global space, denoted as \mathbf{v} , are determined by the following motion rules:

$$\mathbf{v} = LBS(\mathbf{v}_{base} + BS(\{\beta; \epsilon; \psi\}; W_{bs}), J(\psi), W), \quad (10)$$

where \mathbf{v}_{base} represents the vertices of the FLAME mesh in canonical space. *LBS* signifies the standard Linear Blend Skinning function with weights W and the function J stands for the joint regressor based on the pose ψ , both as defined in the FLAME model. The *BS* operation denotes a linear blend shaping process that creates facial animations by combining blendshapes in accordance with FLAME parameters, weighted by the blendshape weights W_{bs} . For a triangle \mathcal{T} of the FLAME mesh, which is defined by its vertices $\{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2\}$ and its edges $\{\mathbf{e}_{01} = \mathbf{v}_1 - \mathbf{v}_0, \mathbf{e}_{02} = \mathbf{v}_2 - \mathbf{v}_0, \mathbf{e}_{12} = \mathbf{v}_2 - \mathbf{v}_1\}$, we establish a local coordinate system based on \mathcal{T} . The system’s origin, P , representing \mathcal{T} ’s position in global space, is determined by a barycentric combination of the triangle’s vertices. The system’s orientation in global space is captured by the rotation matrix R , and the scale factor S reflects the size of \mathcal{T} . These components are delineated as follows:

$$P = \eta_0 \mathbf{v}_0 + \eta_1 \mathbf{v}_1 + \eta_2 \mathbf{v}_2, \quad (11)$$

$$R = [\mathbf{n}_0; \mathbf{n}_1; \mathbf{n}_2], \quad (12)$$

$$S = (\|\mathbf{e}_{01}\| + \|\mathbf{e}_{02}\| + \|\mathbf{e}_{12}\|)/3, \quad (13)$$

where $\boldsymbol{\eta} = (\eta_0, \eta_1, \eta_2)$ is the learnable barycentric coefficients, and the orthogonal unit vectors $[\mathbf{n}_0; \mathbf{n}_1; \mathbf{n}_2]$ are calculated as follows:

$$\mathbf{n}_0 = \frac{\mathbf{e}_{01}}{\|\mathbf{e}_{01}\|}, \mathbf{n}_1 = \frac{\mathbf{e}_{01} \times \mathbf{e}_{02}}{\|\mathbf{e}_{01} \times \mathbf{e}_{02}\|}, \mathbf{n}_2 = \mathbf{n}_0 \times \mathbf{n}_1. \quad (14)$$

Then a Gaussian is associated within this local space, denoted as $\mathcal{G} = \{\tilde{\mathbf{u}}, \tilde{\mathbf{r}}, \tilde{\mathbf{s}}, \alpha, \kappa, \mathcal{T}, \boldsymbol{\eta}\}$, enabling \mathcal{G} to track the movements of \mathcal{T} . Specifically, the attributes $\tilde{\mathbf{u}}, \tilde{\mathbf{r}}, \tilde{\mathbf{s}}$ of \mathcal{G} are defined relative to the local coordinate system, rather than the global space. We initialize its position $\tilde{\mathbf{u}}$ at the origin, rotation $\tilde{\mathbf{r}}$ as the identity matrix, and

scale $\tilde{\mathbf{s}}$ as a unit vector. When rendering, we transform the \mathcal{G} into global space by:

$$\mathbf{u} = R \cdot \tilde{\mathbf{u}} + P, \mathbf{r} = R \cdot \tilde{\mathbf{r}}, \mathbf{s} = S\tilde{\mathbf{s}}, \quad (15)$$

where $\mathbf{u}, \mathbf{r}, \mathbf{s}$ denotes the global position, orientation, scale of \mathcal{G} .

During optimization, we employ an adaptive density control strategy similar to the one described by 3D GS[23], to dynamically add or remove Gaussians based on the view space positional gradient and the opacity of each Gaussian. Specifically, when a Gaussian \mathcal{G}_i is anchored to a triangle \mathcal{T}_i of the FLAME mesh, all its derivatives generated by cloning or splitting from \mathcal{G}_i , denoted as $(\mathcal{G}_{i,1}, \mathcal{G}_{i,2}, \mathcal{G}_{i,3}, \dots)$, inherit the same local coordinate system that we established based on \mathcal{T}_i .

3.3.2 Speaker-specific BlendShapes. We have addressed the challenge of capturing expressive facial motion at a coarse level by binding Gaussians to the triangles of the FLAME mesh. However, since FLAME primarily encodes only coarse facial deformations, it remains a challenge to reproduce speaker-specific facial geometry and texture details, such as teeth and wrinkles. To overcome this limitation, we introduce a set of learnable BlendShape (BS) weights that are designed to capture and delineate these nuanced facial features. Initially, we define a Multilayer Perceptron (MLP), designated as W_γ , which accepts the pose parameters ψ from the FLAME and generates a latent pose representation $\boldsymbol{\gamma}$ as follows:

$$\boldsymbol{\gamma} = W_\gamma(\psi). \quad (16)$$

To refine geometrical details, specialized BS weights, W_{pos} for position and W_{rot} for rotation, are employed to update the corresponding attributes of Gaussian \mathcal{G} within the local space:

$$\tilde{\mathbf{u}}' = \tilde{\mathbf{u}} + BS(\boldsymbol{\gamma}; W_{pos}), \tilde{\mathbf{r}}' = \tilde{\mathbf{r}} \cdot BS(\boldsymbol{\gamma}; W_{rot}). \quad (17)$$

These revised local attributes, $\tilde{\mathbf{u}}'$ and $\tilde{\mathbf{r}}'$, are then converted to global space when rendering, as explained in Equation 15. Similarly, for texture refinement, we apply compensation to κ_0 , the zeroth-order coefficient of κ in \mathcal{G} , which governs the base color calculated by zeroth-order spherical harmonics function. This adjustment is achieved through BS weights designated as W_{color} :

$$\kappa'_0 = \kappa_0 + BS(\boldsymbol{\gamma}; W_{color}). \quad (18)$$

With the incorporation of Speaker-specific BlendShapes for facial detailing, \mathcal{G} is represented as $\mathcal{G} = \{\tilde{\mathbf{u}}', \tilde{\mathbf{r}}', \tilde{\mathbf{s}}, \alpha, \kappa'_0, \kappa_{rest}, \mathcal{T}, \boldsymbol{\eta}\}$, where κ_{rest} denotes the higher-order(1 to k -th) coefficients of κ .

3.3.3 Inpainting Generator. By incorporating Dynamic Deformation and Speaker-specific BlendShapes, we achieve a consistent alignment of the head pose with the original face image, which enables a stable reintegration of the rendered face into the original video frame. Nonetheless, using an alternative audio source may lead to discrepancies in lip shapes between the rendered and original video frames. These discrepancies are particularly pronounced around the facial contour and chin, resulting in visual inconsistencies. To address these issues and enhance the integrity of the composite image, we introduce an Inpainting Generator, denoted as F . It is designed to seamlessly fill in these mismatches and improve the visual continuity of the final frames. The blending process is described by the following equation:

$$I = (1 - M) \cdot I_{ori} + M \cdot F(I_{gau} + (1 - M)I_{ori}), \quad (19)$$

Table 1: Quantitative results under the self-driven setting. The best and second-best results are in bold and underlined.

Methods	PSNR \uparrow	SSIM \uparrow	LPIPS \downarrow	FID \downarrow	LMD \downarrow	LSE-C \uparrow	LSE-D \downarrow	FPS \uparrow
Ground Truth	—	—	—	—	—	7.465	7.131	—
Wav2Lip[36]	<u>33.6435</u>	<u>0.9506</u>	<u>0.0602</u>	17.44	6.048	9.129*	5.860*	23
AD-NeRF[17]	30.7756	0.9200	0.1135	26.98	3.975	6.128	8.134	0.15
RAD-NeRF[45]	29.9283	0.9115	0.1075	18.73	<u>3.519</u>	6.096	8.147	37
ER-NeRF[26]	29.5774	0.9071	0.0694	13.22	3.555	6.340	7.955	38
GeneFace++[56]	27.4192	0.8870	0.0920	<u>12.69</u>	3.942	5.920	8.378	<u>44</u>
Ours	37.0775	0.9676	0.0239	4.57	3.278	<u>7.015</u>	<u>7.562</u>	130

where M denotes the facial mask extracted from the original frame by face parsing[62], and I_{ori} and I_{gau} denote the original video frame and the output image of the Gaussian renderer, respectively.

3.3.4 Training Object. While training the Dynamic Gaussian Renderer, we focus on three types of optimization objectives: image similarity, Gaussian attributes, and Gaussian semantic category.

Image Similarity. The image similarity loss comprises a combination of L1 loss \mathcal{L}_1 , VGG loss \mathcal{L}_{vgg} , and SSIM loss \mathcal{L}_{ssim} :

$$\mathcal{L}_{rgb} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_{vgg} + \lambda_3 \mathcal{L}_{ssim}. \quad (20)$$

Gaussian Attributes. We aim to ensure that during optimization, the positions of the Gaussians faithfully align with the associated FLAME triangles and their sizes remain within reasonable bounds to avoid visual jitter and artifacts. To achieve this, we apply constraints on the Gaussians' position and scale:

$$\mathcal{L}_{attr} = \lambda_p \|\max(0, \mathbf{u}' - \epsilon_p)\|^2 + \lambda_s \|\max(0, \bar{s} - \epsilon_s)\|^2, \quad (21)$$

where ϵ_p and ϵ_s serve as predefined thresholds that set the maximum allowable positions for \mathbf{u}' and scales for \bar{s} , respectively.

Gaussian Semantic Category. The dynamic deformation aims for the deformed Gaussians to approximate the associated FLAME mesh. Ideally, Gaussians bound to specific triangles, like the lips, should exactly match the corresponding region's color and position in the rendered image. However, it's a challenge to prevent Gaussians from shifting to implausible positions during optimization, resulting in jittering artifacts. To address this issue, we introduce a Gaussian semantic loss \mathcal{L}_{seg} . We render an auxiliary semantic segmentation map M_{aux} using the Gaussian Renderer, where instead of utilizing the learned color \mathbf{c}_i in Equation 9, we assign a fixed color \mathbf{c}_{fix} to each Gaussian point based on its parent triangle's category; for example, assigning red to facial points and yellow to lip points:

$$C_{seg} = \sum_{i=1} \mathbf{c}_{fix} \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j). \quad (22)$$

Then, we compute the mean square error between M_{gau} and M_{gt} obtained by face parsing[62]:

$$\mathcal{L}_{seg} = \lambda_{seg} \|M_{aux} - M_{gt}\|^2. \quad (23)$$

Consequently, the final loss function is formulated as follows:

$$\mathcal{L} = \mathcal{L}_{rgb} + \mathcal{L}_{attr} + \mathcal{L}_{seg}. \quad (24)$$

* Wav2Lip is jointly trained with SyncNet, and LSE-C and LSE-D are its optimization objectives. As a result, it obtains better scores than the ground truth.

Table 2: Quantitative results under the cross-driven setting. The best and second-best results are in bold and underlined.

Methods	Testset A			Testset B		
	FID \downarrow	LSE-C \uparrow	LSE-D \downarrow	FID \downarrow	LSE-C \uparrow	LSE-D \downarrow
Wav2Lip[36]	18.34	8.380	6.985	17.18	9.106	6.479
AD-NeRF[17]	29.95	3.929	11.02	29.92	3.408	11.39
RAD-NeRF[45]	19.55	4.028	10.78	19.18	4.035	10.93
ER-NeRF[26]	13.68	4.780	10.12	13.40	4.680	10.26
GeneFace++[56]	<u>12.77</u>	5.734	9.030	<u>12.92</u>	6.040	8.759
Ours	5.136	<u>5.758</u>	<u>8.665</u>	4.394	<u>7.148</u>	<u>7.617</u>

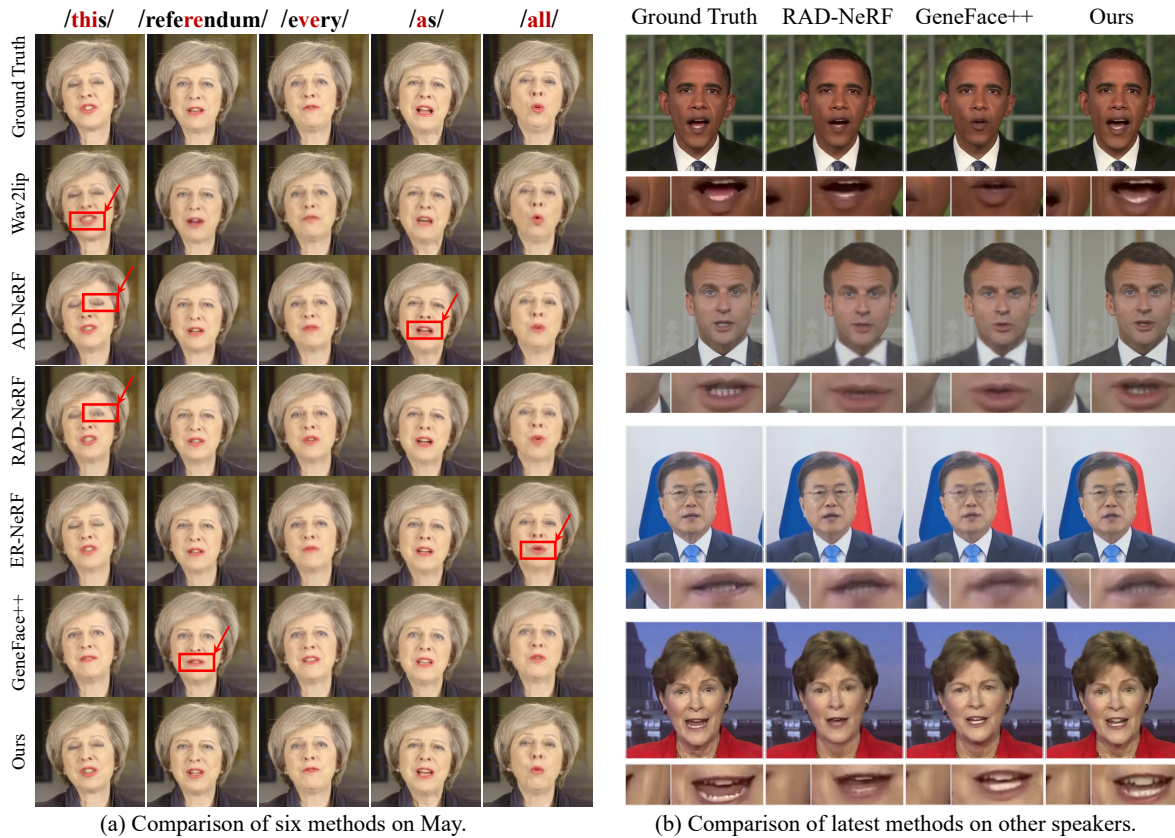
4 EXPERIMENT

4.1 Experimental Settings

4.1.1 Datasets. To facilitate comparison with other state-of-the-art methods, we utilize the same dataset as AD-NeRF and GeneFace++, comprised of five videos averaging 9085 frames each, at a frame rate of 25 FPS, across English, French, and Korean languages. To improve the Speaker-specific Motion Translator's generalization capabilities, we incorporate data from target speakers, supplemented with 100 samples of Chinese CN-CVS videos[8] and 100 samples of English HDTF videos[61] for joint training. During the data preprocessing phase, we begin by extracting audio and video frames from the source video. Subsequently, facial keypoints are extracted and smoothed[5, 25], then employ EMOCA[13] to obtain FLAME parameters sequence. Finally, we apply the face parsing technique[62], which segments each video frame into distinct facial regions.

4.1.2 Compared baselines. We compare our method with several representative methods: 1) Wav2Lip[36]: Sync-expert supervised learning enhances lip sync accuracy; 2) AD-NeRF[17]: NeRF synthesizes audio-driven videos end-to-end; 3) RAD-NeRF[45]: Discrete learnable mesh boosts rendering efficiency; 4) ER-NeRF[26]: Three-plane hash representation enables high-quality rendering; 5) GeneFace++[56]: Pitch-aware audio-to-motion module and landmark LLE method ensure stable rendering.

4.1.3 Implementation Details. The official FLAME model lacks vertices and triangles for teeth, so we manually attach it with 262 vertices and 546 faces to depict the teeth and inner mouth regions. These newly added elements are not assigned independent BS and LBS weights but rather are designed to move in sync with the



(a) Comparison of six methods on May.

(b) Comparison of latest methods on other speakers.

Figure 4: The comparison of generated key frame results. We show the ground truth frames for comparing and mark the un-sync and bad rendering quality results with red arrows. Please zoom in for better visualization.

lip area of the original FLAME model. For GaussianTalker’s training, we employ the Adam Optimizer[24] across all modules. The Speaker-specific Motion Translator is trained for 100,000 iterations, with the batch size set to 1. This training takes about 6 hours, using a learning rate of 1×10^{-4} . Furthermore, we train the Dynamic Gaussian Renderer for 50,000 iterations using a batch size of 8, and this training lasts about 4 hours. While optimizing the Gaussians’ attributes, we fine-tune the FLAME parameters within the dataset to ensure they more accurately reflect the face of each frame. All experiments are performed on a single NVIDIA RTX4090 GPU.

4.2 Quantitative Evaluation

4.2.1 Comparison settings. In quantitative evaluation, we focus on both synthesized quality of the head and accuracy of lip movements. Our comparisons are divided into two settings: 1) Self-driven setting, where each speaker’s own audio is used to drive the corresponding reference frame. 2) Cross-driven setting, where we extract two extra audio clips from HDTF[61], named **Testset A** and **Testset B**, to drive each speaker. For each generated result, we rescaled frames into the same size for a fair comparison.

4.2.2 Evaluation Metrics. We employ Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) for assessing

overall image quality. To assess image details, we utilize Learned Perceptual Image Patch Similarity (LPIPS)[60]. To quantify the similarity between the generated video and the ground truth, we employ Fréchet Inception Distance (FID)[18]. For assessing audio-lip synchronization, we utilize Landmark Distance (LMD)[9] to gauge the extent of synchronization with the ground truth, and complementarily evaluate lip movement accuracy through Lip Sync Error Confidence (LSE-C) and Lip Sync Error Distance (LSE-D)[36].

4.2.3 Evaluation Results. The results are presented in Table 1. Our results indicate that, in comparison with leading audio-driven talking head video synthesis methods, our method outperforms others, achieving the highest scores across all image quality assessment metrics. Owing to measures implemented to reduce jitter and artifacts, the videos produced by our method more accurately resemble the actual scenes. Regarding lip synchronization, although Wav2Lip excels in the LSE-C and LSE-D metrics owing to its joint training with SyncNet, our method still surpasses others in these metrics. Additionally, our method demonstrates superior performance in the LMD metric, suggesting a closer and more accurate alignment with actual lip movements.

We also compared the generalization ability of each method using FID score and LSE-C and LSE-D metrics with out-of-distribution (OOD) audio inputs, detailed in Table 2. The results show that our

method exhibits outstanding image quality performance, with FID reflecting a considerable enhancement over competing methods. Furthermore, we have significantly improved the model’s generalization capabilities by training the Speaker-specific Motion Translator on a multi-individual dataset. This enhancement enables our method to attain superior performance in lip synchronization, as evidenced by the LSE-C and LSE-D.

We also tested the rendering speed. With an NVIDIA RTX4090 GPU and data preloaded into memory, a sequence of video frames can be output at 130 FPS. The output video has the same resolution as the original video. This far exceeds the performance of other methods. In addition, we developed a rendering pipeline for Gaussians with MNN[21] and OpenGL[53] and subsequently deployed GaussianTalker. On devices with the Apple M1 chip, it achieved rendering speeds of 36 FPS, demonstrating its adaptability across various platforms.

4.3 Qualitative Evaluation

To more effectively assess image quality and lip synchronization, we present a comparative analysis of our method with others in Figure 4. Compared to our method, Wav2Lip primarily concentrates on lip synchronization yet suffers low image quality, particularly around the mouth area. AD-NeRF and RAD-NeRF cannot naturally perform the blinking process, as evidenced by artifacts appearing in eye closure frames. Moreover, these two methods suffer from head-torso separation when driving cross-identity audio. ER-NeRF demonstrates noticeable jittering in both the head and torso during speech, which severely impacts the realism of the video. GeneFace++ shows limited expressiveness and lip movement variety, failing to perform actions like staring or frowning convincingly.

Most NeRF-based approaches train the head and torso independently, which can compromise the torso’s rendering quality in scenes with extensive torso movement. Furthermore, rendering can introduce artifacts stemming from the incomplete separation of the speaker from the background during preprocessing. In contrast, our approach not only delivers superior image clarity but also achieves enhanced lip synchronization performance, particularly with cross-identity and cross-gender audio inputs, showcasing robust generalization capabilities. Please see our supplementary for better visualization.

4.4 Ablation study

In this section, we conducted ablation studies in a self-driven setting to validate the effectiveness of each component. Quantitative assessment metrics included PSNR, LPIPS, and LMD. The results are shown in Table 3 and Figure 5.

4.4.1 Universal Audio Encoder. We investigate the impact of audio decoupling in the audio encoder, which excludes identity information from the original audio and extracts audio feature that contains only content information. Removing audio decoupling decreases all metrics, with the LMD metric suffering the most, indicating poorer lip synchronization. Figure 5 (a) also illustrates the artifact issues in some of the lip shapes.

4.4.2 Speaker-specific BlendShapes. We explore the impact of eliminating the Speaker-specific BlendShapes, which play a crucial role

Table 3: Ablation study on subject Obama. The best overall results are in bold.

Ablation	PSNR \uparrow	LPIPS \downarrow	LMD \downarrow
full	38.37	0.0089	3.72
replace Universal Audio Encoder with Wav2Vec 2.0[54]	36.91	0.0104	4.55
w/o Speaker-specific BlendShapes	37.46	0.0098	3.78
w/o Fine-tune FLAME Parameters	36.87	0.0106	4.28
w/o Gaussian Semantic Loss	37.09	0.0104	3.97

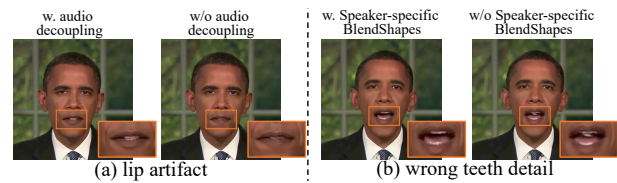


Figure 5: Ablation study on audio decoupling and Speaker-specific BlendShape. Removing them will lead to (a) and (b).

in depicting facial details. The absence may cause the Gaussian renderer to generate artifacts in regions that are not explicitly governed by FLAME parameters, like teeth and wrinkles, resulting in distortions, particularly in fine features, as shown in Figure 5 (b).

4.4.3 Fine-tune FLAME Parameters. This optimization aligns the FLAME parameters with the original video and reduces inter-frame jitter. Without it, the Gaussian point cloud learns biased facial representations. As a result, the visual jitter issue becomes more pronounced and the image quality metrics are significantly declined.

4.4.4 Gaussian Semantic Loss. We investigate the effect of ablating Gaussian semantic loss, which clarifies the binding relationship between 3D Gaussians and FLAME and further normalizes the motion of Gaussians. Without the Gaussian semantic loss, the speaker will jitter slightly, leading to a reduction in image quality metrics.

5 CONCLUSION

In this work, we propose GaussianTalker, a novel framework for audio-driven talking head synthesis via 3D Gaussian Splatting integrated with the FLAME model. GaussianTalker associates multimodal data with specific speakers, reducing potential identity bias between audio, 3D mesh, and video. The Speaker-specific FLAME Translator employs identity decoupling and personalized embedding to achieve synchronized and natural lip movements, while the Dynamic Gaussian Renderer refines Gaussian attributes through a latent pose for stable and realistic rendering. Extensive experiments showed that GaussianTalker outperforms state-of-the-art performance in talking head synthesis, while achieving ultra-high rendering speed that significantly surpasses other methods. We believe that this innovative approach will encourage future studies to develop more fluid and lifelike character expressions and movements. By leveraging advanced Gaussian models and generative techniques, the animation of characters will go far beyond mere lip-syncing, capturing a broader range of character dynamics.

REFERENCES

- [1] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. 2018. LRS3-TED: a large-scale dataset for visual speech recognition. *arXiv preprint arXiv:1809.00496* (2018).
- [2] Volker Blanz and Thomas Vetter. 2023. A morphable model for the synthesis of 3D faces. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 157–164.
- [3] Matthew Brand. 1999. Voice puppetry. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. 21–28.
- [4] Christoph Bregler, Michele Covell, and Malcolm Slaney. 2023. Video rewrite: Driving visual speech with audio. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 715–722.
- [5] Adrian Bulat and Georgios Tzimiropoulos. 2017. How far are we from solving the 2D & 3D Face Alignment problem? (and a dataset of 230,000 3D facial landmarks). In *International Conference on Computer Vision*.
- [6] Chen Cao, Yanlin Weng, Shun Zhou, Yiyang Tong, and Kun Zhou. 2013. Faceware-house: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics* 20, 3 (2013), 413–425.
- [7] Aggelina Chatziagiapi, ShahRukh Athar, Abhinav Jain, MV Rohith, Vimal Bhat, and Dimitris Samaras. 2023. LipNeRF: What is the right feature space to lip-sync a NeRF?. In *2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG)*. IEEE, 1–8.
- [8] Chen Chen, Dong Wang, and Thomas Fang Zheng. 2023. CN-CVS: A mandarin audio-visual dataset for large vocabulary continuous visual to speech synthesis. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
- [9] Lele Chen, Zhiheng Li, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. 2018. Lip Movements Generation at a Glance. In *Proceedings of the European Conference on Computer Vision (ECCV)*.
- [10] Lele Chen, Ross K Maddox, Zhiyao Duan, and Chenliang Xu. 2019. Hierarchical cross-modal talking face generation with dynamic pixel-wise loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 7832–7841.
- [11] Yufan Chen, Lizhen Wang, Qijing Li, Hongjiang Xiao, Shengping Zhang, Hongxun Yao, and Yebin Liu. 2023. Monogaussianavatar: Monocular gaussian point-based head avatar. *arXiv preprint arXiv:2312.04558* (2023).
- [12] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. 2018. Voxceleb2: Deep speaker recognition. *arXiv preprint arXiv:1806.05622* (2018).
- [13] Radek Daněček, Michael J Black, and Timo Bolkart. 2022. Emoca: Emotion driven monocular face capture and animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20311–20322.
- [14] Tony Ezzat, Gadi Geiger, and Tomaso Poggio. 2002. Trainable videorealistic speech animation. *ACM Transactions on Graphics (TOG)* 21, 3 (2002), 388–398.
- [15] Guy Gafni, Justus Thies, Michael Zollhofer, and Matthias Nießner. 2021. Dynamic neural radiance fields for monocular 4d facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8649–8658.
- [16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).
- [17] Yudong Guo, Keyu Chen, Sen Liang, Yong-Jin Liu, Hujun Bao, and Juyong Zhang. 2021. Ad-nerf: Audio driven neural radiance fields for talking head synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5784–5794.
- [18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (Long Beach, California, USA) (NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6629–6640.
- [19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.
- [20] Amir Jamaludin, Joon Son Chung, and Andrew Zisserman. 2019. You said that?: Synthesising talking faces from audio. *International Journal of Computer Vision* 127 (2019), 1767–1779.
- [21] Xiaotang Jiang, Huan Wang, Yiliu Chen, Ziqi Wu, Lichuan Wang, Bin Zou, Yafeng Yang, Zongyang Cui, Yu Cai, Tianhang Yu, Chengfei Lv, and Zhihua Wu. 2020. MNN: A Universal and Efficient Inference Engine. In *MLSys*.
- [22] Tero Karras, Timo Aila, Samuli Laine, Antti Herva, and Jaakko Lehtinen. 2017. Audio-driven facial animation by joint end-to-end learning of pose and emotion. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 1–12.
- [23] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. (2023).
- [24] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [25] Till Kroeger, Radu Timofte, Dengxin Dai, and Luc Van Gool. 2016. Fast Optical Flow using Dense Inverse Search. *arXiv:arXiv:1603.03590*
- [26] Jiahe Li, Jiawei Zhang, Xiao Bai, Jun Zhou, and Lin Gu. 2023. Efficient region-aware neural radiance fields for high-fidelity talking portrait synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7568–7578.
- [27] Tianye Li, Timo Bolkart, Michael J Black, Hao Li, and Javier Romero. 2017. Learning a model of facial shape and expression from 4D scans. *ACM Trans. Graph.* 36, 6 (2017), 194–1.
- [28] Xueting Li, Shalini De Mello, Sifei Liu, Koki Nagano, Umar Iqbal, and Jan Kautz. 2024. Generalizable One-shot 3D Neural Head Avatar. *Advances in Neural Information Processing Systems* 36 (2024).
- [29] Xian Liu, Yinghao Xu, Qianyi Wu, Hang Zhou, Wayne Wu, and Bolei Zhou. 2022. Semantic-aware implicit neural audio-driven video portrait generation. In *European conference on computer vision*. Springer, 106–125.
- [30] Yuanxun Lu, Jinxiang Chai, and Xun Cao. 2021. Live speech portraits: real-time photorealistic talking-head animation. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–17.
- [31] B Mildenhall, PP Srinivasan, M Tancik, JT Barron, R Ramamoorthi, and R Ng. 2020. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*.
- [32] Shigeo Morishima. 1998. Real-time Talking Head Driven by Voice and its Application to Communication and Entertainment. 195–200.
- [33] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. 2021. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5865–5874.
- [34] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. 2009. A 3D face model for pose and illumination invariant face recognition. In *2009 sixth IEEE international conference on advanced video and signal based surveillance*. Ieee, 296–301.
- [35] Ziqiao Peng, Yihao Luo, Yue Shi, Hao Xu, Xiangyu Zhu, Hongyan Liu, Jun He, and Zhaoxin Fan. 2023. Selftalk: A self-supervised commutative training diagram to comprehend 3d talking faces. In *Proceedings of the 31st ACM International Conference on Multimedia*. 5292–5301.
- [36] KR Prajwal, Rudrabha Mukhopadhyay, Vinay P Nambodiri, and CV Jawahar. 2020. 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*. 484–492.
- [37] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. 2021. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10318–10327.
- [38] Zengyi Qin, Wenliang Zhao, Xumin Yu, and Xin Sun. 2023. OpenVoice: Versatile Instant Voice Cloning. *arXiv preprint arXiv:2312.01479* (2023).
- [39] Alexander Richard, Michael Zollhöfer, Yandong Wen, Fernando De la Torre, and Yaser Sheikh. 2021. Meshtalk: 3d face animation from speech using cross-modality disentanglement. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1173–1182.
- [40] Alfredo Rivero, ShahRukh Athar, Zhixin Shu, and Dimitris Samaras. 2024. Rig3DGS: Creating Controllable Portraits from Casual Monocular Videos. *arXiv preprint arXiv:2402.03723* (2024).
- [41] Jiayang Shang, Tianwei Shen, Shiwei Li, Lei Zhou, Mingmin Zhen, Tian Fang, and Long Quan. 2020. Self-supervised monocular 3d face reconstruction by occlusion-aware multi-view geometry consistency. In *European Conference on Computer Vision*. Springer, 53–70.
- [42] Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. 2024. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. *arXiv preprint arXiv:2403.05087* (2024).
- [43] Shuai Shen, Wanhua Li, Zheng Zhu, Yueqi Duan, Jie Zhou, and Jiwen Lu. 2022. Learning dynamic facial radiance fields for few-shot talking head synthesis. In *European conference on computer vision*. Springer, 666–682.
- [44] Hiroki Tanaka and Satoshi Nakamura. 2022. The Acceptability of Virtual Characters as Social Skills Trainers: Usability Study. *JMIR Hum Factors* 9, 1 (29 Mar 2022), e35358. <https://doi.org/10.2196/35358>
- [45] Jiayang Tang, Kaisiyuan Wang, Hang Zhou, Xiaokang Chen, Dongliang He, Tianshu Hu, Jingtuo Liu, Gang Zeng, and Jingdong Wang. 2022. Real-time neural radiance talking portrait synthesis via audio-spatial decomposition. *arXiv preprint arXiv:2211.12368* (2022).
- [46] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. 2020. Neural voice puppetry: Audio-driven facial reenactment. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*. Springer, 716–731.
- [47] Justus Thies, Mohamed Elgharib, Ayush Tewari, Christian Theobalt, and Matthias Nießner. 2020. Neural Voice Puppetry: Audio-Driven Facial Reenactment. In *Computer Vision – ECCV 2020*, Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (Eds.). Springer International Publishing, Cham, 716–731.
- [48] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. *CoRR* abs/1807.03748 (2018). [arXiv:1807.03748](http://arxiv.org/abs/1807.03748) <http://arxiv.org/abs/1807.03748>
- [49] Konstantinos Vougioukas, Stavros Petridis, and Maja Pantic. 2020. Realistic speech-driven facial animation with gans. *International Journal of Computer*

929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044

- 1045 *Vision* 128, 5 (2020), 1398–1413.
- 1046 [50] Jie Wang, Jiu-Cheng Xie, Xianyan Li, Feng Xu, Chi-Man Pun, and Hao Gao. 2023. GaussianHead: Impressive Head Avatars with Learnable Gaussian Diffusion. *arXiv preprint arXiv:2312.01632* (2023).
- 1047
- 1048 [51] Kaisiyuan Wang, Qianyi Wu, Linsen Song, Zhuoqian Yang, Wayne Wu, Chen
- 1049 Qian, Ran He, Yu Qiao, and Chen Change Loy. 2020. Mead: A large-scale audio-
- 1050 visual dataset for emotional talking-face generation. In *European Conference on*
- 1051 *Computer Vision*. Springer, 700–717.
- 1052 [52] Olivia Wiles, A Koepke, and Andrew Zisserman. 2018. X2face: A network for
- 1053 controlling face generation using images, audio, and pose codes. In *Proceedings*
- 1054 *of the European conference on computer vision (ECCV)*. 670–686.
- 1055 [53] Mason Woo, Jackie Neider, Tom Davis, and Dave Shreiner. 1999. *OpenGL pro-*
- 1056 *gramming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley
- 1057 Longman Publishing Co., Inc.
- 1058 [54] Qiantong Xu, Alexei Baevski, and Michael Auli. 2021. Simple and Effective
- 1059 Zero-shot Cross-lingual Phoneme Recognition. *arXiv:2109.11680* [cs.CL]
- 1060 [55] Shunyu Yao, RuiZhe Zhong, Yichao Yan, Guangtao Zhai, and Xiaokang Yang. 2022.
- 1061 Dfa-nerf: Personalized talking head generation via disentangled face attributes
- 1062 neural rendering. *arXiv preprint arXiv:2201.00791* (2022).
- 1063 [56] Zhenhui Ye, Jinzheng He, Ziyue Jiang, Rongjie Huang, Jiawei Huang, Jinglin Liu,
- 1064 Yi Ren, Xiang Yin, Zejun Ma, and Zhou Zhao. 2023. GeneFace++: Generalized
- 1065 and Stable Real-Time Audio-Driven 3D Talking Face Generation. *arXiv preprint*
- 1066 *arXiv:2305.00787* (2023).
- 1067 [57] Zhenhui Ye, Ziyue Jiang, Yi Ren, Jinglin Liu, Jinzheng He, and Zhou Zhao. 2023.
- 1068 GeneFace: Generalized and high-fidelity audio-driven 3d talking face synthesis.
- 1069 *arXiv preprint arXiv:2301.13430* (2023).
- 1070 [58] Ran Yi, Zipeng Ye, Juyong Zhang, Hujun Bao, and Yong-Jin Liu. 2020. Audio-
- 1071 driven talking face video generation with natural head pose. *arXiv preprint*
- 1072 *arXiv:2002.10137* 2, 6 (2020), 7.
- 1073
- 1074
- 1075
- 1076
- 1077
- 1078
- 1079
- 1080
- 1081
- 1082
- 1083
- 1084
- 1085
- 1086
- 1087
- 1088
- 1089
- 1090
- 1091
- 1092
- 1093
- 1094
- 1095
- 1096
- 1097
- 1098
- 1099
- 1100
- 1101
- 1102
- [59] Lingyun Yu, Jun Yu, Mengyan Li, and Qiang Ling. 2020. Multimodal inputs driven
- 1103 talking face generation with spatial–temporal dependency. *IEEE Transactions on*
- 1104 *Circuits and Systems for Video Technology* 31, 1 (2020), 203–216.
- [60] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang.
- 1105 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric.
- 1106 In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*
- 1107 *(CVPR)*.
- [61] Zhimeng Zhang, Lincheng Li, Yu Ding, and Changjie Fan. 2021. Flow-Guided One-
- 1108 Shot Talking Face Generation With a High-Resolution Audio-Visual Dataset. In
- 1109 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- 1110 3661–3670.
- [62] Yinglin Zheng, Hao Yang, Ting Zhang, Jianmin Bao, Dongdong Chen, Yangyu
- 1111 Huang, Lu Yuan, Dong Chen, Ming Zeng, and Fang Wen. 2022. General facial rep-
- 1112 resentation learning in a visual-linguistic manner. In *Proceedings of the IEEE/CVF*
- 1113 *Conference on Computer Vision and Pattern Recognition*. 18697–18709.
- [63] Hang Zhou, Yu Liu, Ziwei Liu, Ping Luo, and Xiaogang Wang. 2019. Talking
- 1114 face generation by adversarially disentangled audio-visual representation. In
- 1115 *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 9299–9306.
- [64] Yang Zhou, Xintong Han, Eli Shechtman, Jose Echevarria, Evangelos Kalogerakis,
- 1116 and Dingzeyu Li. 2020. Makeltalk: speaker-aware talking-head animation. *ACM*
- 1117 *Transactions On Graphics (TOG)* 39, 6 (2020), 1–15.
- [65] Zhenglin Zhou, Fan Ma, Hehe Fan, and Yi Yang. 2024. HeadStudio: Text to Ani-
- 1118 matable Head Avatars with 3D Gaussian Splatting. *arXiv preprint arXiv:2402.06149*
- 1119 (2024).
- [66] Michael Zollhöfer, Justus Thies, Pablo Garrido, Derek Bradley, Thabo Beeler,
- 1120 Patrick Pérez, Marc Stamminger, Matthias Nießner, and Christian Theobalt. 2018.
- 1121 State of the art on monocular 3D face reconstruction, tracking, and applications.
- 1122 In *Computer graphics forum*, Vol. 37. Wiley Online Library, 523–550.
- 1123
- 1124
- 1125
- 1126
- 1127
- 1128
- 1129
- 1130
- 1131
- 1132
- 1133
- 1134
- 1135
- 1136
- 1137
- 1138
- 1139
- 1140
- 1141
- 1142
- 1143
- 1144
- 1145
- 1146
- 1147
- 1148
- 1149
- 1150
- 1151
- 1152
- 1153
- 1154
- 1155
- 1156
- 1157
- 1158
- 1159
- 1160